A modal specification theory for components with data

Bauer, Sebastian S.; Larsen, Kim G.; Legay, Axel; Nyman, Ulrik; Wasowski, Andrzej

Published in:
Science of Computer Programming

DOI (link to publication from Publisher):
10.1016/j.scico.2013.06.003

Publication date:
2014

Document Version
Accepted author manuscript, peer reviewed version

A Modal Specification Theory for Components with Data*

Sebastian S. Bauer

Institut für Informatik, Ludwig-Maximilians-Universität München, Germany

Kim G. Larsen

Computer Science, Aalborg University, Denmark

Axel Legay

INRIA/IRISA, Rennes Cedex, France & Computer Science, Aalborg University, Denmark

Ulrik Nyman

Computer Science, Aalborg University, Denmark

Andrzej Wąsowski

IT University of Copenhagen, Denmark

Abstract

Modal specification is a well-known formalism used as an abstraction theory for transition systems. Modal specifications are transition systems equipped with two types of transitions: *must*-transitions that are mandatory to any implementation, and *may*-transitions that are optional. The duality of transitions allows for developing a unique approach for both logical and structural compositions, and eases the step-wise refinement process for building implementations. We propose Modal Specifications with Data (MSDs), the first *modal* specification theory with explicit representation of data. Our new theory includes the most commonly seen ingredients of a specification theory; that is parallel composition, conjunction and quotient. As MSDs are by nature potentially infinite-state systems, we propose symbolic representations based on effective predicates. Our theory serves as a new abstraction-based formalism for transition systems with data.

Keywords: Modal Transition Systems, Specifications, Interfaces, Data Abstraction, Modal Specifications with Data.

Email addresses: bauerse@pst.ifi.lmu.de (Sebastian S. Bauer), kgl@cs.aau.dk (Kim G. Larsen), axel.legay@irisa.fr (Axel Legay), ulrik@cs.aau.dk (Ulrik Nyman), wasowski@itu.dk (Andrzej Wąsowski)

*The present paper is an extended version of the conference paper [1] with the semantic definitions, more proofs, an additional figure and extended related work.

1. Introduction

Modern IT systems are often large and consist of complex assemblies of numerous reactive and interacting components. The components are often designed by independent teams, working under a common agreement on what the interface of each component should be. Consequently, the search for mathematical foundations which support *compositional reasoning* on interfaces is a major research goal. A framework should support inferring properties of the global implementation, designing and advisedly reusing components.

Interfaces are specifications and components that implement an interface are understood as models, or implementations. Specification theories should support various features including (1) *refinement*, which allows to compare specifications as well as to replace a specification by another one in a larger design, (2) *structural composition*, which allows to combine specifications of different components, (3) *logical conjunction*, expressing the intersection of the set of requirements expressed by two or more specifications for the same component, and last (4) a *quotient operator* that is dual to structural composition and allows synthesizing a component from a set of assumptions.

Among existing specification theories, one finds modal specifications [2], which are labeled transition systems equipped with two types of transitions: *must*-transitions that are mandatory for any implementation, and *may*-transitions which are optional for an implementation. Modal specifications are known to achieve a more flexible and easy-to-use compositional development methodology for CCS [3], which includes a considerable simplification of the step-wise refinement process proposed by Milner and Larsen. While being very close to logics (conjunction), the formalism takes advantage of a behavioral semantics allowing for easy composition with respect to process construction (structural composition) and synthesis (quotient). However, despite the many advantages, only a few implementations have been considered so far. One major problem is that contrary to other formalisms based on transition systems, there exists no theory of modal specification equipped with rich information such as data variables.

In this paper, we add a new stone to the cathedral of results on modal specifications [4, 5], that is we propose the first such theory equipped with rich data values. Our first contribution is to design a semantical version of modal specifications whose states are split into locations and valuations for possibly infinite-domain variables. For every component, we distinguish between local variables, that are locally controlled by the component, and global uncontrolled variables that are controlled by other components and can be accessed, but not modified. Combining variables with sets of actions labeling transitions offers a powerful set of communication primitives that cannot be captured by most existing specification theories. We also propose a symbolic predicate-based representation of our formalism. We consider effective predicates that are closed under conjunction, union, and membership—classical assumptions in existing symbolic theories (e.g. [6]). While the semantic level is possibly infinite-state, the syntactical level permits us to reason on specifications just like one would with the original modal specifications, but with the additional power of rich data. We see this as the most important contribution of this paper. We see the potential of handling infinite data domains as the most important contribution of this paper. An important direction of future work is to establish case studies where infinite data domains are used extensively.

Continuing our quest, we study modal refinement between specifications. Refinement, which resembles simulation between transition systems, permits to compare sets of implementations in a syntactic manner. Modal refinement is defined at the semantic level, but can also be checked at the symbolic level. We propose a predicate abstraction approach that simplifies the practical complexity of the operation by reducing the number of states and simplifying the predicates. This approach is in line with the work of Godefroid et al. [7], but is applied to specification-based verification rather than to model checking.

We then propose definitions for both logical and structural composition, both on the level of symbolic representations of specifications and on the semantic level. The syntactic definitions are clearly not direct extensions of the ones defined on modal specifications as behaviors of both controlled and uncontrolled variables have to be taken into account. As usual, structural composition offers the property of independent implementability, hence allowing for elegant step-wise refinement. In logical composition, two specifications which disagree on their requirements can be reconciled by synthesizing a new component where conflicts have been removed. This can be done with a symbolic pruning of bad states, which terminates if the system is finite-state, or if the structure of the transition system induced by the specification relies, for instance, on a well-quasi order [8]. Finally, we also propose a quotient operation, that is the dual operation of structural composition, which works for a subclass of systems, and we discuss its limitation. This operator, absent from most existing behavioral and logical specification theories, allows synthesizing a component from a set of assumptions.

This journal paper is an extended version of the conference paper [1]; it contains additional semantic definitions of the relations and operations on the level of MSDs, proofs for all results, more details on predicate abstraction as well as an extended section on related work.

In Sect. 2 we introduce modal specifications with data and their finite symbolic representations, refinement, an implementation relation and consistency. In Sect. 3 we define the essential operators of every specification theory, that is parallel composition, conjunction and quotient. For verification of refinement between infinite-state specifications we propose in Sect. 4 an approach based on predicate abstraction techniques. We summarize related works in Sect. 5 and conclude and discuss future work in Sect. 6.

2. Modal Specifications with Data

We will first introduce specifications which are finite symbolic representations of modal specifications with data (MSDs). We will then propose modal refinement and derive an implementation relation and a consistency notion.

Figure 1 shows the relationship between specifications, MSDs and implementations (TSD which are introduced later). Figure 2 shows the implications that exists between the different forms of refinement that are presented in the paper.

In the following, $\mathscr{P}(M)$ denotes the powerset of M, $\mathscr{P}_{\geq 1}(M) = \mathscr{P}(M) \setminus \{\emptyset\}$, and the union of two disjoint sets is denoted by $M \uplus N$, which is $M \cup N$ with $M \cap N = \emptyset$.

We assume that variables range over a fixed domain $\mathbb{D}$. For a given set V of variables, a *data state* s over V is a mapping $s : V \to \mathbb{D}$. If $V = \{x_1, x_2, \ldots, x_n\}$ and

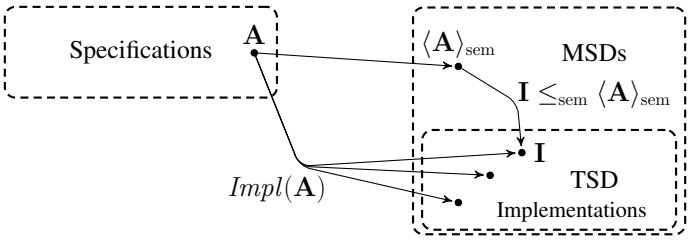


Figure 1: Figure showing the relationship between specifications, MSDs (modal specifications with data) and TSD (implementations).

Specifications		$\mathbf{A} \leq \mathbf{B}$
	$\Updownarrow$	
MSDs		$\langle \mathbf{A} \rangle_{\text{sem}} \leq_{\text{sem}} \langle \mathbf{B} \rangle_{\text{sem}}$
	$\Downarrow$	
TSD (Implementations)		$\mathit{Impl}(\mathbf{A}) \subseteq \mathit{Impl}(\mathbf{B})$

Figure 2: Implications between the two refinement relations and implementation set inclusion.

$d_1, d_2, \ldots, d_n \in \mathbb{D}$, we write $[x_1 \mapsto d_1, x_2 \mapsto d_2, \ldots, x_n \mapsto d_n]$ for the data state s which maps every x_i to d_i, for $1 \leq i \leq n$. We write $[\![V]\!]$ for the set of all possible data states over V. For disjoint sets of variables V_1 and V_2 and data states $s_1 \in [\![V_1]\!]$ and $s_2 \in [\![V_2]\!]$, the operation $(s_1 \cdot s_2)$ composes the data states resulting in a new state $s = (s_1 \cdot s_2) \in [\![V_1 \uplus V_2]\!]$, such that $s(x) = s_1(x)$ for all $x \in V_1$ and $s(x) = s_2(x)$ for all $x \in V_2$. This is naturally lifted to sets of states: if $S_1 \subseteq [\![V_1]\!]$ and $S_2 \subseteq [\![V_2]\!]$ then $(S_1 \cdot S_2) = \{(s_1 \cdot s_2) \mid s_1 \in S_1, s_2 \in S_2\} \subseteq [\![V_1 \uplus V_2]\!]$.

Like in the work of de Alfaro et al. [9] we define specifications with respect to an assertion language allowing suitable predicate representation. Given a set V of variables, we denote by $\mathit{Pred}(V)$ the set of first-order predicates with free variables in V; we assume that these predicates are written in some specified first-order language with existential ($\exists$) and universal ($\forall$) quantifiers and with interpreted function symbols and predicates; in our examples, the language contains the usual arithmetic operators and boolean connectives ($\vee, \wedge, \neg, \Rightarrow$). Given a set of variables V we denote by $(V)'$ an isomorphic set of 'primed' variables from V: so if $x \in V$ then $(x)' \in (V)'$. We use this construction to represent pre- and post-values of variables. A variable $(x)' \in (V)'$ represents the next state value of the variable $x \in V$. Given a formula $\varphi \in \mathit{Pred}(V)$ and a data state $s \in [\![V]\!]$, we write $\varphi(s)$ if the predicate formula φ is true when its free variables are interpreted as specified by s. Given a formula $\psi \in \mathit{Pred}(V_1 \uplus (V_2)')$ and states $s_1 \in [\![V_1]\!]$, $s_2 \in [\![V_2]\!]$, we often write $\psi(s_1, s_2)$ for $\psi(s_1 \cdot t_2)$ where $t_2 \in [\![(V_2)']\!]$ such that $t_2((x)') = s_2(x)$ for all $x \in V_2$. Given a predicate $\varphi \in \mathit{Pred}(V)$, we write $(\varphi)' \in \mathit{Pred}((V)')$ for the predicate obtained by substituting x with $(x)'$ in φ, for all $x \in V$; similarly, for $\varphi \in \mathit{Pred}((V)')$ we write $\varphi\downarrow \in \mathit{Pred}(V)$ for the predicate

obtained by substituting every $(x)' \in (V)'$ with its unprimed version. We write $[\![\varphi]\!]$ for the set $\{s \in [\![V]\!] \mid \varphi(s)\}$ which consists of all states satisfying $\varphi \in \mathit{Pred}(V)$ (for predicates with primed and unprimed variables), and φ is *consistent* if $[\![\varphi]\!] \neq \emptyset$. We write $\exists V \varphi$ meaning existential quantification of φ over all variables in the set V, and similar for universal quantification. Finally, for a predicate $\psi \in \mathit{Pred}(V_1 \uplus (V_2)')$, we write ${}^\circ\psi$ for the pre-projection $\exists (V_2)' \psi$, and ψ° for the post-projection $\exists V_1 \psi$.

Our theory enriches modal transition systems with variables. We use the terms modal specifications and modal transition systems interchangeably throughout the paper. Specifications not only express constraints on the allowed sequences of actions, but also their dependence and effect on the values of variables. Like in the loose approach of modal specifications [2] which allows under-specification using *may* and *must* modalities on transitions, we allow loose specification of the effects of actions on the data state. From a given location and a given data state, a transition to another location is allowed to lead to several next data states.

A *signature* $\mathit{Sig} = (\Sigma, V^L, V^G)$ determines the alphabet of actions Σ and the set of variables $V = V^L \uplus V^G$ of an interface. The variables in V^L are *local (controlled) variables*, owned by the interface and visible to any other component. V^G contains the *global (uncontrolled) variables* owned by the environment, which are read-only for the interface.

Specifications are finite modal transition systems where transitions are equipped with predicates. A transition predicate $\psi \in \mathit{Pred}(V \uplus (V^L)')$ relates a previous state, determined by all controlled and uncontrolled data states, with the next possible controlled data state.

Definition 1. A *specification* is a tuple $\mathbf{A} = (\mathit{Sig}, \mathit{Loc}, \ell^0, \varphi^0, E_\Diamond, E_\Box)$ where $\mathit{Sig} = (\Sigma, V^L, V^G)$ is a signature, Loc is a finite set of locations, $\ell^0 \in \mathit{Loc}$ is the initial location, $\varphi^0 \in \mathit{Pred}(V^L)$ is a predicate on the initial local state, and $E_\Diamond$, $E_\Box$ are finite may- and must-transition relations respectively:

$$E_\Diamond, E_\Box \subseteq \mathit{Loc} \times \Sigma \times \mathit{Pred}(V \uplus (V^L)') \times \mathit{Loc}.$$

Given a specification $\mathbf{A}$, locations $\ell, \ell' \in \mathit{Loc}$, and action $a \in \Sigma$, we refer to the set of transition predicates on may-transitions by $\mathit{May}^a(\ell, \ell') = \{\psi \mid (\ell, a, \psi, \ell') \in E_\Diamond\}$ and on must-transitions by $\mathit{Must}^a(\ell, \ell') = \{\psi \mid (\ell, a, \psi, \ell') \in E_\Box\}$.

Example 1. Consider a specification of a print server, shown in Fig. 3. Must-transitions are drawn with solid arrows and may-transitions with dashed ones. Every solid arrow representing a must-transition has an implicit may-transition shadowing it which is not shown. Every transition is equipped with a transition predicate over unprimed variables, referring to the pre-state, and primed variables, referring to the poststate. The print server receives new print jobs (**newPrintJob**), stores them and assigns them either a low or high priority; the numbers of low and high priority jobs are modeled by controlled variables l and h, respectively; l and h are natural numbers. A job with low priority can also be reclassified to high priority (**incPriority**). The print server can send (**send**) a job to a printer, and then wait for the acknowledgment (**ack**). In state ℓ_1, if there is a job with high priority and the uncontrolled boolean variable $\mathit{priorityMode}$

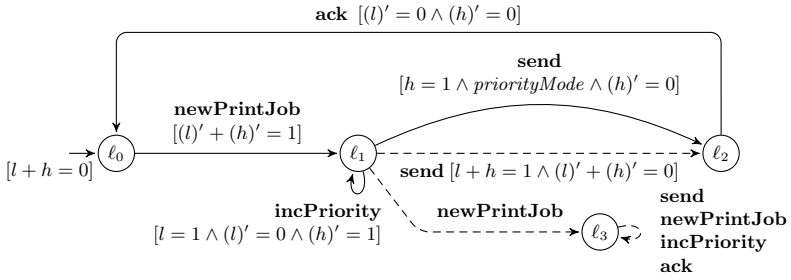


Figure 3: Abstract specification $\mathbf{P}$ of a print server.

is true, then there must be a send transition. The specification is loose in the sense that if a second print job is received in state ℓ_1, then the behavior is left unspecified.

We now define the kind of transition systems which will be used for formalizing the semantics of specifications. A specification is interpreted as a variant of modal transition systems where the *state space* is formed by the cartesian product $Loc \times [\![V^L]\!]$, i.e. a *state* is a pair (ℓ, s) where $\ell \in Loc$ is a location and $s \in [\![V^L]\!]$ is a valuation of the controlled variables [10, 11]. To motivate the choice of the transition relations in the semantics of specifications, we first describe the intended meaning of may- and must-transitions.

A may-transition $(\ell, a, \psi, \ell') \in E_\Diamond$ in the specification expresses that in any implementation, in any state (ℓ, s) and for any guard $g \in [\![V^G]\!]$ (that is a valuation of the global uncontrolled variables V^G) the implementation is *allowed* to have a transition with guard g and action a to a next state (ℓ', s') such that $\psi(s \cdot g, s')$. The interpretation of a must-transition $(\ell, a, \psi, \ell') \in E_\Box$ is a bit more involved: Any implementation, in state (ℓ, s), and for any guard $g \in [\![V^G]\!]$, if there is a valuation $s' \in [\![V^L]\!]$ such that $\psi(s \cdot g, s')$, then the implementation is *required* to have a transition from state (ℓ, s) with guard g and action a to *at least some* state t' such that $\psi(s \cdot g, t')$. The requirement expressed by must-transitions cannot be formalized by standard modal transition systems, but fortunately, a generalization called disjunctive modal transition systems introduced in [12] can precisely capture these requirements. A may-transition targets (as usual) only one state, while a disjunctive must-transitions can branch to several possible next states (thus must-transitions are hypertransitions), with an existential interpretation: there must exist at least one transition with some target state which is an element from the set of target states of the hypertransition.

Definition 2. A *modal specification with data (MSD)* is a tuple

$$\mathbf{S} = (Sig, Loc, \ell^0, S^0, \longrightarrow_\Diamond, \longrightarrow_\Box)$$

where Sig, Loc, ℓ^0 are like in Def. 1, $S^0 \subseteq [\![V^L]\!]$ is a set of initial data states, and $\longrightarrow_\Diamond, \longrightarrow_\Box \subseteq Loc \times [\![V^L]\!] \times [\![V^G]\!] \times \Sigma \times (Loc \times \mathscr{P}_{\geq 1}([\![V^L]\!]))$ are the may- ($\Diamond$) and must- ($\Box$) transition relations such that every may-transition targets a single state: if $(\ell, s, g, a, (\ell', S')) \in \longrightarrow_\Diamond$ then $|S'| = 1$.

A state $(\ell, s) \in Loc \times [\![V^L]\!]$ is called *syntactically consistent* iff targets reachable by must-transitions are also reachable by may-transitions: if $(\ell, s, g, a, (\ell', S')) \in \longrightarrow_\Box$

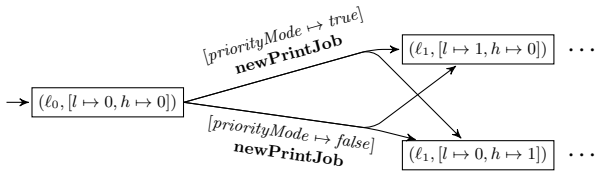


Figure 4: Excerpt of the semantics of the abstract print server specification.

then $(\ell, s, g, a, (\ell', \{s'\})) \in \longrightarrow_\Diamond$ for all $s' \in S'$. $\mathbf{S}$ is *syntactically consistent* iff all states are syntactically consistent, and the set of initial data states is nonempty, i.e. $S^0 \neq \emptyset$.

May-transitions $(\ell, s, g, a, (\ell', S')) \in \longrightarrow_\Diamond$ are often written $(\ell, s) \xrightarrow{g\,a}_\Diamond (\ell', S')$, and similarly for must-transitions.

We can now define formally how a specification translates to its semantics in terms of an MSD. A single may-transition in a specification will give rise to a set of semantic may-transitions pointing to single admissible target states, and a must-transition gives rise to (must-)hypertransitions targeting all the admissible poststates.

Definition 3. The *semantics* of a specification $\mathbf{A} = (Sig, Loc, \ell^0, \varphi^0, E_\Diamond, E_\Box)$ is given by the MSD $\langle \mathbf{A} \rangle_{\text{sem}} = (Sig, Loc, \ell^0, S^0, \longrightarrow_\Diamond, \longrightarrow_\Box)$ where $S^0 = [\![\varphi^0]\!]$ and the transition relations are defined as follows. For each $\ell, \ell' \in Loc$, $s, s' \in [\![V^L]\!]$, $g \in [\![V^G]\!]$, and $a \in \Sigma$:

i. If $(\ell, a, \psi, \ell') \in E_\Diamond$ and $\psi(s \cdot g, s')$ then $(\ell, s) \xrightarrow{g\,a}_\Diamond (\ell', \{s'\})$,

ii. If $(\ell, a, \psi, \ell') \in E_\Box$ and $\psi(s \cdot g, s')$ then $(\ell, s) \xrightarrow{g\,a}_\Box (\ell', \{t' \in [\![V^L]\!] \mid \psi(s \cdot g, t')\})$.

A specification $\mathbf{A}$ is called *MSD consistent* iff its semantics $\langle \mathbf{A} \rangle_{\text{sem}}$ is syntactically consistent. Note that: In the following we will always assume that specifications are MSD consistent and MSDs are syntactically consistent.

Example 2. An excerpt of the semantics of our abstract specification of the print server (see Fig. 3) can be seen Fig. 4. As before, we draw must-transitions with a solid arrow, and have an implicit set of may-transitions shadowing it which are not shown, i.e. for each target (ℓ, S') of a must-transition and each $s \in S'$ there is a may-transition with the same source state and with target state $(\ell, \{s\})$.

The first must-transition $(\ell_0, \mathbf{newPrintJob}, (l)' + (h)' = 1, \ell_1) \in E_\Box$ of the print server specification gives rise to the transitions shown in Fig. 4. Any new print job must be stored in either l or h but which one is not yet fixed by the specification. Thus in the semantics this is expressed as a disjunctive must-transition to the unique location ℓ_1 and the next possible data states $[l \mapsto 1, h \mapsto 0]$ and $[l \mapsto 0, h \mapsto 1]$.

A *refinement relation* allows to relate a concrete specification with an abstract specification. Refinement should satisfy the following substitutability property: If $\mathbf{A}$ refines $\mathbf{B}$ then replacing $\mathbf{B}$ with $\mathbf{A}$ in a context $\mathcal{C}[\cdot]$ gives a specification $\mathcal{C}[\mathbf{A}]$ refining $\mathcal{C}[\mathbf{B}]$. Refinement will be a precongruence, i.e. it is compatible with the structural and logical operators on specifications in the above sense.

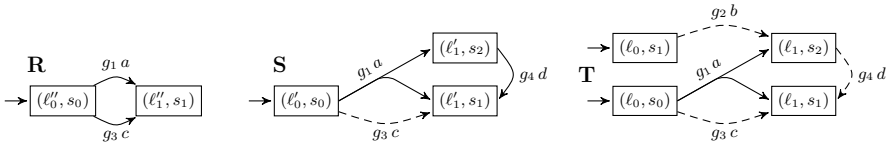


Figure 5: Successive refinement of an MSD $\mathbf{T}$.

Our definition of refinement is based on modal refinement [13, 12] for (disjunctive) modal transition systems, where the may-transitions determine which actions are permitted in a refinement while the must-transitions specify which actions must be present in a refinement and hence in any implementation. We adapt it with respect to data states.

Example 3. We motivate our adaption of modal refinement to take into account data states with the help of a small example shown in Fig. 5. We draw may-transitions with a dashed arrow, and must-transitions with a solid arrow. Every must-transition has an implicit set of may-transitions shadowing it which are not shown. The MSD $\mathbf{T}$ (to the right) has two initial states, both having ℓ_0 as the initial location. The must-transition starting from (ℓ_0, s_0) expresses that in any implementation there must be a transition leading to at least one of the states (ℓ_1, s_1) and (ℓ_1, s_2). The MSD $\mathbf{T}$ can be refined to the MSD $\mathbf{S}$ (by dropping one may-transition and turning one may-transition to a must-transition), and then $\mathbf{S}$ is refined by the MSD $\mathbf{R}$, by refining the must-transition $(\ell_0', s_0, g_1, a, (\ell_1', \{s_1, s_2\}))$ in $\mathbf{S}$ to the must-transition $(\ell_0'', s_0, g_1, a, (\ell_1'', \{s_1\}))$ in $\mathbf{R}$, and by strengthening the transition with guard g_3 and action c to a must-transition.

Definition 4. Let $\mathbf{T}_1 = (Sig, Loc_1, \ell_1^0, S_1^0, \longrightarrow_{\diamond,1}, \longrightarrow_{\square,1})$ and $\mathbf{T}_2 = (Sig, Loc_2, \ell_2^0, S_2^0, \longrightarrow_{\diamond,2}, \longrightarrow_{\square,2})$ be MSDs over the same signature $Sig = (\Sigma, V^L, V^G)$. A relation $R \subseteq Loc_1 \times Loc_2 \times [\![V^L]\!]$ is a *refinement relation* iff for all $(\ell_1, \ell_2, s) \in R$:

i. Whenever $(\ell_1, s) \xrightarrow{g\,a}_{\diamond,1} (\ell_1', \{s'\})$ then there exists $(\ell_2, s) \xrightarrow{g\,a}_{\diamond,2} (\ell_2', \{t'\})$ such that $s' = t'$ and $(\ell_1', \ell_2', s') \in R$.

ii. Whenever $(\ell_2, s) \xrightarrow{g\,a}_{\square,2} (\ell_2', S_2')$ then there exists $(\ell_1, s) \xrightarrow{g\,a}_{\square,1} (\ell_1', S_1')$ such that $S_1' \subseteq S_2'$ and $(\ell_1', \ell_2', s') \in R$ for all $s' \in S_1'$.

We say that $\mathbf{T}_1$ *refines* $\mathbf{T}_2$, written $\mathbf{T}_1 \leq_{\text{sem}} \mathbf{T}_2$, iff $S_1^0 \subseteq S_2^0$ and there exists a refinement relation R such that for any $s \in S_1^0$ also $(\ell_1^0, \ell_2^0, s) \in R$. A specification $\mathbf{A}_1$ refines another specification $\mathbf{A}_2$, written $\mathbf{A}_1 \leq \mathbf{A}_2$, iff $\langle \mathbf{A}_1 \rangle_{\text{sem}} \leq_{\text{sem}} \langle \mathbf{A}_2 \rangle_{\text{sem}}$.

The refinement relation is a preorder on the class of all specifications. Refinement can be checked in polynomial time in the size of the state space of the MSDs (for variables with finite domains). In general the domain may be infinite, or prohibitively large, so in Sect. 4 we revisit the question of refinement checking using abstraction techniques.

Example 4. The semantics of our abstract print server specification, shown in Fig. 4, can be refined as shown in Fig. 6. Now, both must-transitions point to the location ℓ_1 with the data state $[l \mapsto 1, h \mapsto 0]$ which means that any new incoming print job is assigned a low priority, independent of the uncontrolled variable $priorityMode$.

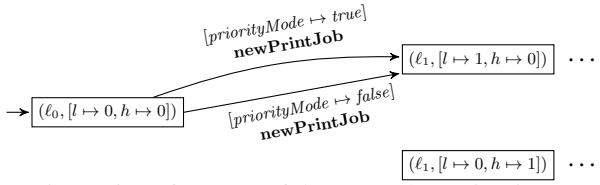


Figure 6: Refinement of the MSD shown in Fig. 4.

An MSD for which the conditions (1) $\longrightarrow_\diamond = \longrightarrow_\Box$ and (2) $|S^0| = 1$ are satisfied, can be interpreted as (an abstraction of) an *implementation*: there are no design choices left open as (1) all may-transitions are covered by must-transitions and (2) there is only one initial data state possible. Any MSD for which the conditions (1) and (2) are satisfied, is called a *transition system with data (TSD)* in the following. Note that TSD cannot be strictly refined, i.e. for any TSD $\mathbf{I}$ and any MSD $\mathbf{S}$ with the same signature, $\mathbf{S} \leq_{\text{sem}} \mathbf{I}$ implies $\mathbf{I} \leq_{\text{sem}} \mathbf{S}$.

An implementation relation connects specifications to implementations (given as TSD) satisfying them. We can simply use refinement as the implementation relation. Given a specification $\mathbf{A}$ and some TSD $\mathbf{I}$, we write $\mathbf{I} \models \mathbf{A}$ for $\mathbf{I} \leq_{\text{sem}} \langle \mathbf{A} \rangle_{\text{sem}}$, so our implementation $\mathbf{I}$ is seen as the model which satisfies the property expressed by the specification $\mathbf{A}$. Now the set of implementations of a specification is the set of all its refining TSD: given a specification $\mathbf{A}$, we define $Impl(\mathbf{A}) = \{\mathbf{I} \mid \mathbf{I} \models \mathbf{A}\}$.

Our implementation relation $\models$ immediately leads to the classical notion of consistency as existence of models. A specification $\mathbf{A}$ is *consistent* iff $Impl(\mathbf{A})$ is non-empty. Consequently, as modal refinement is reflexive, any specification $\mathbf{A}$ for which $\langle \mathbf{A} \rangle_{\text{sem}}$ is a TSD, is consistent. In order to avoid confusion with syntactical consistency of MSDs we have chosen to call consistency of specifications MSD consistency.

By transitivity, modal refinement entails implementation set inclusion: for specifications $\mathbf{A}$ and $\mathbf{B}$, if $\mathbf{A} \leq \mathbf{B}$ then $Impl(\mathbf{A}) \subseteq Impl(\mathbf{B})$. The relation $Impl(\mathbf{A}) \subseteq Impl(\mathbf{B})$ is sometimes called *thorough refinement* [14]. The concept of defining refinement based on implementation set inclusion (known as loose semantics) was first introduced by C.A.R. Hoare [15] in 1972. Just like for modal transition systems, thorough refinement does not imply modal refinement in general [16]. To establish equivalence we follow [17] by imposing a restriction on $\mathbf{B}$, namely that it is deterministic. An MSD is *deterministic* if it is satisfied that

(1) if $(\ell, s, g, a, (\ell', S')), (\ell, s, g, a, (\ell'', S'')) \in \longrightarrow_\Box$ then $\ell' = \ell''$ and $S' = S''$,

(2) if $(\ell, s, g, a, (\ell', \{s'\})), (\ell, s, g, a, (\ell'', \{s''\})) \in \longrightarrow_\diamond$ then $\ell' = \ell''$.

A specification $\mathbf{B}$ is *deterministic*, if the MSD $\langle \mathbf{B} \rangle_{\text{sem}}$ is deterministic. Note that for may-transitions, determinism only requires that for the same source state, guard and action, the transition leads to a unique next location. The reason why this is sufficient is that modal refinement explicitely distinguishes states by their data state part: two states (ℓ', s') and (ℓ'', s'') can only be related if their data state parts s', s'' coincide.

Now, turning back to the relationship of modal refinement and inclusion of implementation sets (thorough refinement), we will prove the following theorem. Under the

restriction of determinism of the refined (abstract) specification we can prove completeness of refinement. This theorem effectively means that modal refinement, as defined for MSDs, is characterized by set inclusion of admitted implementations.

Theorem 1. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two MSD consistent specifications with the same signature such that* $\mathbf{B}$ *is deterministic. Then* $\mathbf{A} \leq \mathbf{B}$ *if and only if* $Impl(\mathbf{A}) \subseteq Impl(\mathbf{B})$.

Proof of Thm. 1. This proof is an adaptation of the proof for completeness of refinement in [17]. Let $\mathbf{S} = \langle \mathbf{A} \rangle_{\text{sem}}$, $\mathbf{T} = \langle \mathbf{B} \rangle_{\text{sem}}$. The implication $\mathbf{S} \leq_{\text{sem}} \mathbf{T} \implies Impl(\mathbf{S}) \subseteq Impl(\mathbf{T})$ immediately follows from transitivity of refinement.

In this proof, we write $(\mathbf{S}, (\ell_{\mathbf{S}}, S))$ for $\mathbf{S}$ where the initial location is replaced with $\ell_{\mathbf{S}}$, and the set of initial data states by S. We can observe that the assumption $Impl(\mathbf{S}) \subseteq Impl(\mathbf{T})$ means more precisely $Impl((\mathbf{S}, (\ell^0_{\mathbf{S}}, S^0_{\mathbf{S}}))) \subseteq Impl((\mathbf{T}, (\ell^0_{\mathbf{T}}, S^0_{\mathbf{T}})))$.

Let $R \subseteq Loc_{\mathbf{S}} \times Loc_{\mathbf{T}} \times [\![V^L]\!]$ be the smallest relation satisfying

- for all $s \in S^0_{\mathbf{S}}$, $(\ell^0_{\mathbf{S}}, \ell^0_{\mathbf{T}}, s) \in R$,
- if $(\ell_{\mathbf{S}}, \ell_{\mathbf{T}}, s) \in R$ and $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{S}} (\ell'_{\mathbf{S}}, \{s'\})$ and $(\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{T}} (\ell'_{\mathbf{T}}, \{s'\})$, then $(\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}, s') \in R$.

We will show that R is a relation witnessing $\mathbf{S} \leq_{\text{sem}} \mathbf{T}$.

First, we prove that $(\ell_{\mathbf{S}}, \ell_{\mathbf{T}}, s) \in R$ implies

$$Impl(\mathbf{S}, (\ell_{\mathbf{S}}, \{s\})) \subseteq Impl(\mathbf{T}, (\ell_{\mathbf{T}}, \{s\})). \tag{1}$$

For $(\ell^0_{\mathbf{S}}, \ell^0_{\mathbf{T}}, s) \in R$, this holds by assumption. Now, assume $(\ell_{\mathbf{S}}, \ell_{\mathbf{T}}, s) \in R$ and $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{S}} (\ell'_{\mathbf{S}}, \{s'\})$ and $(\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{T}} (\ell'_{\mathbf{T}}, \{s'\})$. Let $\mathbf{I}' \in Impl(\mathbf{S}, (\ell'_{\mathbf{S}}, \{s'\}))$. Since $\mathbf{A}$ is MSD consistent we know that $\mathbf{S}$ is syntactically consistent. Then, since $\mathbf{S}$ is syntactically consistent there exists $\mathbf{I} \in Impl(\mathbf{S}, (\ell_{\mathbf{S}}, \{s\}))$ such that $(\ell^0_{\mathbf{I}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{I}} (\ell'_{\mathbf{I}}, \{s'\})$ and $(\mathbf{I}, (\ell'_{\mathbf{I}}, \{s'\})) \leq_{\text{sem}} \mathbf{I}'$, hence also $\mathbf{I}' \leq_{\text{sem}} (\mathbf{I}, (\ell'_{\mathbf{I}}, \{s'\}))$. From (1) it follows that $\mathbf{I} \leq_{\text{sem}} (\mathbf{T}, (\ell_{\mathbf{T}}, \{s\}))$, and since $\mathbf{T}$ is deterministic we can conclude that $(\mathbf{I}, (\ell'_{\mathbf{I}}, \{s'\})) \leq_{\text{sem}} (\mathbf{T}, (\ell'_{\mathbf{T}}, \{s'\}))$, and then $\mathbf{I}' \in Impl(\mathbf{T}, (\ell'_{\mathbf{T}}, \{s'\}))$ by transitivity of refinement.

We now show that R is a relation witnessing $\mathbf{S} \leq_{\text{sem}} \mathbf{T}$. Let $(\ell_{\mathbf{S}}, \ell_{\mathbf{T}}, s) \in R$.

1. Assume $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{S}} (\ell'_{\mathbf{S}}, \{s'\})$. Then there exists an implementation

   $$\mathbf{I} \in Impl(\mathbf{S}, (\ell_S, \{s\}))$$

   such that $(\ell^0_{\mathbf{I}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{I}} (\ell'_{\mathbf{I}}, \{s'\})$. By the assertion above, we know

   $$\mathbf{I} \leq_{\text{sem}} (\mathbf{T}, (\ell_{\mathbf{T}}, \{s\})),$$

   hence there exists $(\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\diamond,\mathbf{T}} (\ell'_{\mathbf{T}}, \{s'\})$. By definition of R, we finally get $(\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}, s') \in R$.

2. Assume $(\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{T}} (\ell'_{\mathbf{T}}, T')$. Then for all $\mathbf{I} \in Impl(\mathbf{T}, (\ell_{\mathbf{T}}, \{s\}))$ there exists $(\ell_{\mathbf{I}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{I}} (\ell'_{\mathbf{I}}, \{s'\})$ for some $s' \in T'$. Since by the above observation (1), $Impl(\mathbf{S}, (\ell_{\mathbf{S}}, \{s\})) \subseteq Impl(\mathbf{T}, (\ell_{\mathbf{T}}, \{s\}))$, we know that every implementation of $(\mathbf{S}, (\ell_{\mathbf{S}}, \{s\}))$ must implement this transition, which implies that there is

a must-transition $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{S}} (\ell'_{\mathbf{S}}, S')$ with $s' \in S'$. We still have to show that $S' \subseteq T'$. To see this, assume $s' \in S' \setminus T'$, then there is an $\mathbf{I} \in Impl(\mathbf{S}, (\ell_{\mathbf{S}}, \{s\}))$ such that $(\ell_{\mathbf{I}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{I}} (\ell'_{\mathbf{I}}, \{s'\})$, and there is no other must-transition with this guard. It also holds that $\mathbf{I} \in Impl(\mathbf{T}, (\ell_{\mathbf{T}}, \{s\}))$, but since T is deterministic, the transition $(\ell_{\mathbf{I}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{I}} (\ell'_{\mathbf{I}}, \{s'\})$ must match with $(\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\Box,\mathbf{T}} (\ell'_{\mathbf{T}}, T')$, hence $s' \in T'$ and this contradicts our assumption. Thus $S' \subseteq T'$, and by definition of R, $(\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}, s')$ for each $s' \in S'$; this follows from the fact that there exist underlying may-transitions in $\mathbf{S}$ and $\mathbf{T}$, respectively, which allows us to reach every $s' \in S'$.

□

Thus having proved that our refinement is thorough we move on to defining and proving theorems about: Parallel composition, pruning and logical composition.

3. Compositional Reasoning

In this section we propose all the essential operators on specifications a good specification theory should provide. We will distinguish between structural and logical composition. Structural composition mimics the classical composition of transition systems at the specification level. Logical composition allows to compute the intersection of sets of models and hence can be used to represent the conjunction of requirements made on an implementation. Furthermore we will introduce a quotient operator which is the dual operator to structural composition.

From now on, we assume that for any two specifications with the signatures $Sig_1 = (\Sigma_1, V_1^L, V_1^G)$ and $Sig_2 = (\Sigma_2, V_2^L, V_2^G)$, respectively, we can assume that $\Sigma_1 = \Sigma_2$ and $V_1^L \uplus V_1^G = V_2^L \uplus V_2^G$. This is not a limitation, as one can apply the constructions of [5] to equalize alphabets of actions and sets of variables.

Parallel composition. Two specifications $\mathbf{A}_1$ and $\mathbf{A}_2$ with $Sig_1 = (\Sigma, V_1^L, V_1^G)$ and $Sig_2 = (\Sigma, V_2^L, V_2^G)$, respectively, are *composable* iff $V_1^L \cap V_2^L = \emptyset$. Then their signatures can be composed in a straightforward manner to the signature

$$Sig_1 \times Sig_2 =_{def} (\Sigma, V_1^L \uplus V_2^L, (V_1^G \cup V_2^G) \setminus (V_1^L \uplus V_2^L))$$

in which the set of controlled variables is the disjoint union of the sets of controlled variables of $\mathbf{A}_1$ and $\mathbf{A}_2$, and the set of uncontrolled variables consists of all those uncontrolled variables of $\mathbf{A}_1$ and $\mathbf{A}_2$ which are controlled neither by $\mathbf{A}_1$ nor by $\mathbf{A}_2$.

Definition 5. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two composable specifications. The *parallel composition* of $\mathbf{A}_1$ and $\mathbf{A}_2$ is defined as the specification

$$\mathbf{A}_1 \parallel \mathbf{A}_2 = (Sig_1 \times Sig_2, Loc_1 \times Loc_2, (\ell_1^0, \ell_2^0), \varphi_1^0 \wedge \varphi_2^0, E_\Diamond, E_\Box)$$

where the transition relations $E_\Diamond$ and $E_\Box$ are the smallest relations satisfying the rules:

1. if $(\ell_1, a, \psi_1, \ell'_1) \in E_{\Diamond,1}$ and $(\ell_2, a, \psi_2, \ell'_2) \in E_{\Diamond,2}$ then $((\ell_1, \ell_2), a, \psi_1 \wedge \psi_2, (\ell'_1, \ell'_2)) \in E_\Diamond$,

2. if $(\ell_1, a, \psi_1, \ell_1') \in E_{\Box,1}$ and $(\ell_2, a, \psi_2, \ell_2') \in E_{\Box,2}$ then $((\ell_1, \ell_2), a, \psi_1 \wedge \psi_2, (\ell_1', \ell_2')) \in E_{\Box}$.

We will also define the parallel composition of two specifications at the semantic level and prove that the symbolic notion of parallel composition is identical to the semantical.

The *parallel composition* of two composable MSDs $\mathbf{S}_1$ and $\mathbf{S}_2$ is defined as the MSD

$$\mathbf{S}_1 \parallel_{\text{sem}} \mathbf{S}_2 = (Sig, Loc_1 \times Loc_2, (\ell_1^0, \ell_2^0), (S_1^0 \cdot S_2^0), \longrightarrow_{\diamond}, \longrightarrow_{\Box})$$

where Sig is like in Def. 5 and where the transition relations are the smallest relations satisfying the rules

$$\frac{(\ell_1, s_1) \xrightarrow{(g \cdot s_2)\, a}_{\diamond,1} (\ell_1', S_1') \qquad (\ell_2, s_2) \xrightarrow{(g \cdot s_1)\, a}_{\diamond,2} (\ell_2', S_2')}{((\ell_1, \ell_2), (s_1 \cdot s_2)) \xrightarrow{g\, a}_{\diamond} ((\ell_1', \ell_2'), (S_1' \cdot S_2'))} [\mathsf{may}_{\parallel}]$$

$$\frac{(\ell_1, s_1) \xrightarrow{(g \cdot s_2)\, a}_{\Box,1} (\ell_1', S_1') \qquad (\ell_2, s_2) \xrightarrow{(g \cdot s_1)\, a}_{\Box,2} (\ell_2', S_2')}{((\ell_1, \ell_2), (s_1 \cdot s_2)) \xrightarrow{g\, a}_{\Box} ((\ell_1', \ell_2'), (S_1' \cdot S_2'))} [\mathsf{must}_{\parallel}]$$

The following theorem characterizes the relation between syntactic and semantic parallel composition.

Theorem 2. *Let $\mathbf{A}_1$, $\mathbf{A}_2$ be two composable specifications. Then $\langle \mathbf{A}_1 \parallel \mathbf{A}_2 \rangle_{\text{sem}} = \langle \mathbf{A}_1 \rangle_{\text{sem}} \parallel_{\text{sem}} \langle \mathbf{A}_2 \rangle_{\text{sem}}$.*

Proof of Thm. 2. In order to prove $\langle \mathbf{A}_1 \parallel \mathbf{A}_2 \rangle_{\text{sem}} = \langle \mathbf{A}_1 \rangle_{\text{sem}} \parallel_{\text{sem}} \langle \mathbf{A}_2 \rangle_{\text{sem}}$, we will show that a must-transition is in $\langle \mathbf{A}_1 \parallel \mathbf{A}_2 \rangle_{\text{sem}}$ if and only if it is in $\langle \mathbf{A}_1 \rangle_{\text{sem}} \parallel_{\text{sem}} \langle \mathbf{A}_2 \rangle_{\text{sem}}$. The proof for may-transitions in similar and is not included here.

Consider a must-transition

$$((\ell_1, \ell_2), s_1 \cdot s_2) \xrightarrow{g\, a}_{\Box, \langle \mathbf{A}_1 \parallel \mathbf{A}_2 \rangle_{\text{sem}}} ((\ell_1', \ell_2'), S') \in \langle \mathbf{A}_1 \parallel \mathbf{A}_2 \rangle_{\text{sem}}$$

Then there exists a transition $((\ell_1, \ell_2), a, \psi, (\ell_1', \ell_2')) \in E_{\Box, \mathbf{A}_1 \parallel \mathbf{A}_2}$ such that $S' = \{s_1' \cdot s_2' \mid \psi(g \cdot s_1 \cdot s_2, s_1' \cdot s_2')\}$. This means that there exist the two transitions

$$(\ell_1, a, \psi_1, \ell_1') \in E_{\Box, \mathbf{A}_1} \text{ and } (\ell_2, a, \psi_2, \ell_2') \in E_{\Box, \mathbf{A}_2}$$

such that $S' = S_1' \cdot S_2'$ where $S_1' = \{s_1' \in [\![V_1^L]\!] \mid \psi_1(g \cdot s_1 \cdot s_2, s_1')\}$ and $S_2' = \{s_2' \in [\![V_2^L]\!] \mid \psi_2(g \cdot s_1 \cdot s_2, s_2')\}$, and $\psi \equiv \psi_1 \wedge \psi_2$. From this we get the two MSD transitions

$$(\ell_1, s_1) \xrightarrow{(g \cdot s_2)\, a}_{\Box, \langle \mathbf{A}_1 \rangle_{\text{sem}}} (\ell_1', S_1') \text{ and } (\ell_2, s_2) \xrightarrow{(g \cdot s_1)\, a}_{\Box, \langle \mathbf{A}_2 \rangle_{\text{sem}}} (\ell_2', S_2')$$

which implies (by the rules of parallel composition)

$$((\ell_1, \ell_2), s_1 \cdot s_2) \xrightarrow{g\, a}_{\Box, \langle \mathbf{A}_1 \rangle_{\text{sem}} \parallel_{\text{sem}} \langle \mathbf{A}_2 \rangle_{\text{sem}}} ((\ell_1', \ell_2'), S_1' \cdot S_2').$$

All the above implications are in fact equivalences, thus every must-transition in $\langle \mathbf{A}_1 \rangle_{\text{sem}} \parallel_{\text{sem}} \langle \mathbf{A}_2 \rangle_{\text{sem}}$ is also a must-transition in $\langle \mathbf{A}_1 \parallel \mathbf{A}_2 \rangle_{\text{sem}}$. □

Composition of specifications, similar to the classical notion of modal composition for modal transition systems [13], synchronizes on matching shared actions and only yields a must-transition if there exist corresponding matching must-transitions in the original specifications. Composition is commutative (up to isomorphism) and associative. Our theory supports independent implementability of specifications, which is a crucial requirement for any compositional specification framework [18].

Theorem 3. *Let* $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1, \mathbf{B}_2$ *be specifications such that* $\mathbf{A}_2$ *and* $\mathbf{B}_2$ *are composable. If* $\mathbf{A}_1 \leq \mathbf{A}_2$ *and* $\mathbf{B}_1 \leq \mathbf{B}_2$*, then* $\mathbf{A}_1 \parallel \mathbf{B}_1 \leq \mathbf{A}_2 \parallel \mathbf{B}_2$.

Proof of Thm. 3. By Thm. 2 it suffices to prove the claim for MSDs $\mathbf{S}', \mathbf{S}, \mathbf{T}', \mathbf{T}$. Assume a relation R_1 proving $\mathbf{S}' \leq_{\text{sem}} \mathbf{S}$ and a relation R_2 which demonstrates $\mathbf{T}' \leq_{\text{sem}} \mathbf{T}$. We show that the following relation R demonstrates $\mathbf{S}' \parallel_{\text{sem}} \mathbf{T}' \leq_{\text{sem}} \mathbf{S} \parallel_{\text{sem}} \mathbf{T}$:

$$R = \{((\ell_{\mathbf{S}'}, \ell_{\mathbf{T}'}), (\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s \cdot t) \mid (\ell_{\mathbf{S}'}, \ell_{\mathbf{S}}, s) \in R_1, (\ell_{\mathbf{T}'}, \ell_{\mathbf{T}}, t) \in R_2\}$$

Here we show the proof for may-transitions, for must-transitions the proof is analogous.

Assume

$$((\ell_{\mathbf{S}'}, \ell_{\mathbf{T}'}), s \cdot t) \xrightarrow{g\, a}_{\diamond, \mathbf{S}' \parallel_{\text{sem}} \mathbf{T}'} ((\ell'_{\mathbf{S}'}, \ell'_{\mathbf{T}'}), \{s' \cdot t'\}).$$

Then, by the rules of parallel composition, we have

$$(\ell_{\mathbf{S}'}, s) \xrightarrow{(g \cdot t)\, a}_{\diamond, \mathbf{S}'} (\ell_{\mathbf{S}'}, \{s'\}) \text{ and } (\ell_{\mathbf{T}'}, t) \xrightarrow{(g \cdot s)\, a}_{\diamond, \mathbf{T}'} (\ell_{\mathbf{T}'}, \{t'\}).$$

By assumption, $(\ell_{\mathbf{S}'}, \ell_{\mathbf{S}}, s) \in R_1$ and $(\ell_{\mathbf{T}'}, \ell_{\mathbf{T}}, t) \in R_2$, implying that there exist

$$(\ell_{\mathbf{S}}, s) \xrightarrow{(g \cdot t)\, a}_{\diamond, \mathbf{S}} (\ell'_{\mathbf{S}}, \{s'\}) \text{ and } (\ell_{\mathbf{T}}, t) \xrightarrow{(g \cdot s)\, a}_{\diamond, \mathbf{T}} (\ell'_{\mathbf{T}}, \{t'\})$$

such that $(\ell'_{\mathbf{S}'}, \ell'_{\mathbf{S}}, s') \in R_1$, $(\ell'_{\mathbf{T}'}, \ell'_{\mathbf{T}}, t') \in R_2$, thus $((\ell_{\mathbf{S}'}, \ell_{\mathbf{T}'}), (\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s' \cdot t') \in R$. □

Remark 1. Interface theories based on transition systems labeled with input/output actions usually involve a notion of compatibility, which is a relation between interfaces determining whether two components can work properly together. Since the present theory does not have a notion of input/output it is enough to require that two components are composable, i.e. that their local variables do not overlap. A pessimistic input/output compatibility notion has been proposed in previous work [19]. Optimistic input/output compatibility based on a game semantics allows computing all the environments in which two components can work together. Following our recent works in [20, 5], one can enrich labels of transitions in the present theory with input and output and apply the same game-based semantics in order to achieve an optimistic composition.

Syntactical consistency. Our next two specification operators, conjunction and quotient, may yield specifications which are *syntactically inconsistent*, i.e. either there is no legal initial data state or there are states with a must-transition but without corresponding may-transition.

In general, given a specification $\mathbf{A}$, MSD consistency implies classical consistency, i.e. $Impl(\mathbf{A}) \neq \emptyset$, but in general, the reverse does not hold. However, every consistent specification can be "pruned" to a MSD consistent one, by pruning backwards from all MSD inconsistent states, removing states which are required to reach some of the "bad" states through must-transitions. Pruning will be shown to preserve the set of implementations.

For a specification $\mathbf{A} = (Sig, Loc, \ell^0, \varphi^0, E_\Diamond, E_\Box)$, the pruning of $\mathbf{A}$, denoted by $\rho(\mathbf{A})$, is done as follows. Let $B : Loc \to Pred(V^L)$ be a mapping of locations to predicates over the local variables. We define a predecessor operation, iteratively computing all states that are forced to reach a "bad" state. Define a weakest precondition predicate, for $\psi \in Pred(V \uplus (V^L)')$, $\varphi \in Pred(V^L)$, by

$$\mathsf{wp}_\psi[\varphi] =_{def} \exists V^G.^\circ\psi \wedge (\forall (V^L)'.\psi \Rightarrow (\varphi)')$$

which computes the largest set of local states such that there exists an uncontrolled state $g \in [\![V^G]\!]$ such that ψ maps to at least one next state, and all next states satisfy φ. Then

$$\mathsf{predec}(B)(\ell) =_{def} B(\ell) \vee \bigvee\nolimits_{a\in\Sigma,\ell'\in Loc,\psi\in Must^a(\ell,\ell')} \mathsf{wp}_\psi[B(\ell')]$$

and $\mathsf{predec}^0(B) =_{def} B$, $\mathsf{predec}^{j+1}(B) =_{def} \mathsf{predec}(\mathsf{predec}^j(B))$ for $j \geq 0$, and $\mathsf{predec}^*(B) =_{def} \bigcup_{j\geq 0} \mathsf{predec}^j(B)$. Define $\mathsf{bad} : Loc \to Pred(V^L)$, for $\ell \in Loc$, by

$$\mathsf{bad}(\ell) =_{def} \bigvee_{a\in\Sigma,\ell'\in Loc,\psi\in Must^a(\ell,\ell')} \exists V^G.^\circ\psi \wedge \left(\forall (V^L)'.\psi \Rightarrow \bigwedge_{\psi'\in May^a(\ell,\ell')} \neg\psi'\right)$$

and thus $\mathsf{bad}(\ell)$ is satisfied by a valuation $s \in [\![V^L]\!]$ iff there is a must-transition for which no choice of the next data state is permitted by the may-transitions.

In general, for infinite-domain variables, the computation of $\mathsf{predec}^*(\mathsf{bad})$ may not terminate. In [8], it was shown that reachability and related properties in well-structured transition systems with data values, that are monotonic transition systems with a well-quasi ordering on the set of data values, is decidable. This result can be used for specifications with infinite-domain variables to show that under these assumptions, there is some $j \geq 0$ such that for all $\ell \in Loc$, $[\![\mathsf{predec}^j(\mathsf{bad})(\ell)]\!] = [\![\mathsf{predec}^{j+1}(\mathsf{bad})(\ell)]\!]$. In the following, for the specification operators conjunction and quotient (which may result in a syntactically inconsistent specification and hence need to be pruned) we assume that such a $j \geq 0$ exists.

The *pruning* $\rho(\mathbf{A})$ of $\mathbf{A}$ is defined if $\varphi^0 \wedge \neg\mathsf{predec}^j(\mathsf{bad})(\ell^0)$ is satisfiable; and in this case, $\rho(\mathbf{A})$ is the specification $(Sig, Loc, \ell^0, \varphi^0 \wedge \neg\mathsf{predec}^j(\mathsf{bad})(\ell^0), E^\rho_\Diamond, E^\rho_\Box)$ where, for $\chi_{bad} = \mathsf{predec}^j(\mathsf{bad})$,

$$E^\rho_\Diamond = \{(\ell_1, a, \neg\chi_{bad}(\ell_1) \wedge \psi \wedge \neg(\chi_{bad}(\ell_2))', \ell_2) \mid (\ell_1, a, \psi, \ell_2) \in E_\Diamond\},$$
$$E^\rho_\Box = \{(\ell_1, a, \neg\chi_{bad}(\ell_1) \wedge \psi \wedge \neg(\chi_{bad}(\ell_2))', \ell_2) \mid (\ell_1, a, \psi, \ell_2) \in E_\Box\}.$$

Crucially the pruning operator has the expected properties:

Theorem 4. *Let* $\mathbf{A}$ *be a deterministic, possibly MSD inconsistent specification. Then* $\rho(\mathbf{A})$ *is defined if and only if* $\mathbf{A}$ *is consistent. And if* $\rho(\mathbf{A})$ *is defined, then*

1. $\rho(\mathbf{A})$ *is a MSD consistent specification,*
2. $\rho(\mathbf{A}) \leq \mathbf{A}$*,*
3. $Impl(\mathbf{A}) = Impl(\rho(\mathbf{A}))$*, and*
4. *for any MSD consistent specification* $\mathbf{B}$*, if* $\mathbf{B} \leq \mathbf{A}$*, then* $\mathbf{B} \leq \rho(\mathbf{A})$*.*

Proof of Thm. 4. First we sketch the proof of the initial claim. That $\rho(\mathbf{A})$ is defined if and only if $\mathbf{A}$ is consistent.
Assume that $\rho(\mathbf{A})$ is not defined, then for any $s \in [\![V^L]\!]$ such that $\varphi^0(s)$, we have that $\mathsf{predec}^*(\mathsf{bad})(\ell^0)(s)$. We will show by induction on $j \geq 0$ that for any state $s \in [\![V^L]\!]$ and any location $\ell \in Loc$,

$$s \in [\![\mathsf{predec}^j(\mathsf{bad})(\ell)]\!] \text{ implies } Impl(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell, \{s\})) = \emptyset$$

where $(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell, \{s\}))$ is $\langle \mathbf{A} \rangle_{\mathrm{sem}}$ in which ℓ^0 is replaced by ℓ and S^0 by $\{s\}$.

For the base case $j = 0$, observe that any state (ℓ, s) for which the data state $s \in [\![V^L]\!]$ satisfies $\mathsf{predec}^0(\mathsf{bad})(\ell) = \mathsf{bad}(\ell)$, that is

$$\bigvee_{a \in \Sigma, \ell' \in Loc, \psi \in Must^a(\ell,\ell')} \exists V^G.^\circ \psi \wedge \left(\forall (V^L)'.\psi \Rightarrow \bigwedge_{\psi' \in May^a(\ell,\ell')} \neg \psi' \right),$$

cannot be implemented, i.e. $Impl(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell, \{s\})) = \emptyset$, because there is a must-transition enabled for which there is either no may-transition at all (the empty conjunction is *true*) or there are may-transitions but there is no legal next data state.

For the induction step, $j > 0$, we assume

$$s \in [\![\mathsf{predec}^j(\mathsf{bad})(\ell^0)]\!] = [\![\mathsf{predec}(\mathsf{predec}^{j-1}(\mathsf{bad}(\ell^0)))]\!].$$

This means that either $s \in [\![\mathsf{predec}^{j-1}(\mathsf{bad}(\ell^0))]\!]$ or s satisfies

$$\bigvee_{a \in \Sigma, \ell' \in Loc, \psi \in Must^a(\ell,\ell')} \mathsf{wp}_\psi[\mathsf{predec}^{j-1}(\mathsf{bad}(\ell'))].$$

In the first case, by the induction hypothesis, it follows that $Impl(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell, \{s\})) = \emptyset$. In the second case, this means that there exists $(\ell, a, \psi, \ell') \in E_\Box$ and $g \in [\![V^G]\!]$ and $s' \in [\![V^L]\!]$ such that $\psi(s{\cdot}g, s')$, and for all $s' \in [\![V^L]\!]$, whenever $\psi(s{\cdot}g, s')$ then s' satisfies $\mathsf{predec}^{j-1}(\mathsf{bad}(\ell'))$; again by the induction hypothesis, $Impl(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell', \{s'\})) = \emptyset$, and thus, any implementation in $Impl(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell, \{s\}))$ must implement the must-transition $(\ell, a, \psi, \ell') \in E_\Box$ which necessarily leads to a state $(\ell', \{s'\})$ for which there cannot exist an implementation. Thus $Impl(\langle \mathbf{A} \rangle_{\mathrm{sem}}, (\ell, \{s\})) = \emptyset$.

We briefly sketch the other direction. If $\rho(A)$ is defined, then we can easily define an implementation of $\mathbf{A}$, by refining all must-transitions to lead to a data state which is not satisfying the predicate $\mathsf{predec}^*(\mathsf{bad})(\ell)$ for the current location ℓ, which is possible (otherwise this state would have been pruned).

In the following we also sketch the proofs for the numbered claims of the theorem.

1. $\rho(\mathbf{A})$ is trivially syntactically consistent because of the definition of the pruned transition relations $E^\rho_\Diamond, E^\rho_\Box$ and the fact that the initial state predicate

$$\varphi^0 \wedge \neg \mathsf{predec}^*(\mathsf{bad})(\ell^0)$$

is consistent.

2. It can be easily shown that the relation

$$R = \{(\ell, \ell, s) \mid \ell \in \mathit{Loc}, s \in [\![V^L]\!] \setminus [\![\mathsf{predec}^*(\mathsf{bad})(\ell)]\!]\}$$

is a relation witnessing $\rho(\mathbf{A}) \leq \mathbf{A}$.

3. The inclusion of implementations $\mathit{Impl}(\rho(\mathbf{A})) \subseteq \mathit{Impl}(\mathbf{A})$ follows from the fact that $\rho(\mathbf{A}) \leq \mathbf{A}$. Let $\mathbf{I} \in \mathit{Impl}(\mathbf{A})$, then there is a relation witnessing $\mathbf{I} \leq_{\mathrm{sem}} \langle \mathbf{A} \rangle_{\mathrm{sem}}$, and now it is straightforward to show that the same relation also witnesses $\mathbf{I} \leq_{\mathrm{sem}} \langle \rho(\mathbf{A}) \rangle_{\mathrm{sem}}$; note that the relation cannot contain any inconsistent states (i.e. having no implementations) because this would contradict with the fact that $\mathbf{I}$ is an implementation.

4. The same argumentation as in the previous point also applies here. (Except for the note about inconsistent states.)

□

In the following we define pruning at the semantic level of MSD and prove that it is equivalent to pruning as defined for specifications.

For an MSD $\mathbf{S} = (\mathit{Sig}, \mathit{Loc}, \ell^0, S^0, \longrightarrow_\Diamond, \longrightarrow_\Box)$, the pruning of $\mathbf{S}$, denoted by $\rho_{\mathrm{sem}}(\mathbf{S})$, is done as follows. Let $B \subseteq (\mathit{Loc} \times [\![V^L]\!])$ be a subset of its states. We define a predecessor operation, iteratively computing all states that are forced to reach a set B of "bad" states.

$$\begin{aligned}
\mathsf{predec}(B) &= \{(\ell, s) \in (\mathit{Loc} \times [\![V^L]\!]) \mid \ (\ell, s, g, a, (\ell', S')) \in \longrightarrow_\Box \text{ such that} \\
&\qquad\qquad \text{for all } s' \in S' : (\ell', s') \in B\} \\
\mathsf{predec}^0(B) &= B \\
\mathsf{predec}^{j+1}(B) &= \mathsf{predec}(\mathsf{predec}^j(B)) \text{ for } j \geq 0 \\
\mathsf{predec}^*(B) &= \bigcup_{j \geq 0} \mathsf{predec}^j(B)
\end{aligned}$$

The *pruning* $\rho_{\mathrm{sem}}(\mathbf{S})$ of $\mathbf{S}$ is defined iff $\{s \in S^0 \mid (\ell^0, s) \notin \mathsf{predec}^*(\mathsf{bad})\} \neq \emptyset$ where $\mathsf{bad} = \{(\ell, s) \in (\mathit{Loc} \times [\![V^L]\!]) \mid (\ell, s)$ immediately syntactically inconsistent $\}$, and in this case, $\rho_{\mathrm{sem}}(\mathbf{S})$ is the syntactically consistent MSD

$$(\mathit{Sig}, \mathit{Loc}, \ell^0, S^0 \cap ([\![V^L]\!] \setminus \mathsf{predec}^*(\mathsf{bad})), \longrightarrow^\rho_\Diamond, \longrightarrow^\rho_\Box)$$

where

$$\begin{aligned}
\longrightarrow^\rho_\Diamond &= \big\{(\ell, s, g, a, (\ell', \{s'\})) \in \longrightarrow_\Diamond \mid (\ell, s) \notin \mathsf{predec}^*(\mathsf{bad}), (\ell', s') \notin \mathsf{predec}^*(\mathsf{bad})\big\} \\
\longrightarrow^\rho_\Box &= \big\{(\ell, s, g, a, (\ell', S')) \in \longrightarrow_\Box \mid (\ell, s) \notin \mathsf{predec}^*(\mathsf{bad}), \\
&\qquad\qquad \text{for all } s' \in S' : (\ell', s') \notin \varphi^*(\mathsf{bad})\big\}
\end{aligned}$$

The pruning on the level of MSDs is in fact equivalent to the pruning on the level of specifications.

Theorem 5. *For any (possibly syntactically inconsistent) specification* $\mathbf{A}$*, it holds that* $\langle \rho(\mathbf{A}) \rangle_{\mathrm{sem}} = \rho_{\mathrm{sem}}(\langle \mathbf{A} \rangle_{\mathrm{sem}})$.

The proof of Thm. 5 is not very difficult and is therefore omitted.

Logical composition. Conjunction of two specifications yields the greatest lower bound with respect to modal refinement. Syntactic inconsistencies arise if one specification requires a behavior disallowed by the other.

Definition 6. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two specifications with the same signature $Sig = (\Sigma, V^L, V^G)$. The *conjunction of* $\mathbf{A}_1$ *and* $\mathbf{A}_2$ is defined as the possibly syntactically inconsistent specification

$$\mathbf{A}_1 \wedge \mathbf{A}_2 = (Sig, Loc_1 \times Loc_2, (\ell_1^0, \ell_2^0), \varphi_1^0 \wedge \varphi_2^0, E_\Diamond, E_\Box)$$

where the transition relations $E_\Diamond$, $E_\Box$ are the smallest relations satisfying the rules, for any $\ell_1, \ell_1' \in Loc_1$, $\ell_2, \ell_2' \in Loc_2$, $a \in \Sigma$,

1. If $(\ell_1, a, \psi_1, \ell_1') \in E_{\Diamond,1}$, $(\ell_2, a, \psi_2, \ell_2') \in E_{\Diamond,2}$, then $((\ell_1, \ell_2), a, \psi_1 \wedge \psi_2, (\ell_1', \ell_2')) \in E_\Diamond$,
2. If $(\ell_1, a, \psi_1, \ell_1') \in E_{\Box,1}$, then $((\ell_1, \ell_2), a, \psi_1 \wedge (\bigvee_{\psi_2 \in May_2^a(\ell_2, \ell_2')} \psi_2), (\ell_1', \ell_2')) \in E_\Box$,
3. If $(\ell_2, a, \psi_2, \ell_2') \in E_{\Box,2}$, then $((\ell_1, \ell_2), a, \psi_2 \wedge (\bigvee_{\psi_1 \in May_1^a(\ell_1, \ell_1')} \psi_1), (\ell_1', \ell_2')) \in E_\Box$,
4. If $(\ell_1, a, \psi_1, \ell_1') \in E_{\Box,1}$ then $((\ell_1, \ell_2), a, {}^\circ\psi_1 \wedge \left(\forall (V^L)'.\psi_1 \Rightarrow \bigwedge_{\psi_2 \in M} \neg\psi_2\right), (\ell_1, \ell_2)) \in E_\Box$, where $M = \bigcup_{\ell_2' \in Loc_2} May_2^a(\ell_2, \ell_2')$,
5. If $(\ell_2, a, \psi_2, \ell_2') \in E_{\Box,2}$ then $((\ell_1, \ell_2), a, {}^\circ\psi_2 \wedge \left(\forall (V^L)'.\psi_2 \Rightarrow \bigwedge_{\psi_1 \in M} \neg\psi_1\right), (\ell_1, \ell_2)) \in E_\Box$, where $M = \bigcup_{\ell_1' \in Loc_1} May_1^a(\ell_1, \ell_1')$.

The first rule composes may-transitions (with the same action) by conjoining their predicates. Rule (2) expresses that any required behavior of $\mathbf{A}_1$, as long as it is allowed by $\mathbf{A}_2$, is also a required behavior in $\mathbf{A}_1 \wedge \mathbf{A}_2$. Rule (3) is identical but with $\mathbf{A}_1$ and $\mathbf{A}_2$ swapped. Rule (4) captures the case when a required behavior of $\mathbf{A}_1$ is not allowed by $\mathbf{A}_2$. Again rule (5) is identical but with $\mathbf{A}_1$ and $\mathbf{A}_2$ swapped. There are two ways in which the required behavior of $\mathbf{A}_1$ can be dis-allowed by $\mathbf{A}_2$: either there is no may-transition at all enabled (left part of the formulas), or the local next states specified by ψ_2 implies the negation of every next local states of the may transitions in $\mathbf{A}_2$.

Let $\mathbf{S}_1 = (Sig, Loc_1, \ell_1^0, S_1^0, \longrightarrow_{\diamond,1}, \longrightarrow_{\Box,1})$ and $\mathbf{S}_2 = (Sig, Loc_2, \ell_2^0, S_2^0, \longrightarrow_{\diamond,2}, \longrightarrow_{\Box,2})$ be two MSDs. The *conjunction of* $\mathbf{S}_1$ *and* $\mathbf{S}_2$ is defined as the possibly syntactically inconsistent MSD

$$\mathbf{S}_1 \wedge_{\mathrm{sem}} \mathbf{S}_2 = (Sig, Loc_1 \times Loc_2, (\ell_1^0, \ell_2^0), S_1^0 \cap S_2^0, \longrightarrow_{\diamond}, \longrightarrow_{\Box})$$

where the transition relations are the smallest relations satisfying, for all $\ell_1, \ell_1' \in Loc_1$, $\ell_2, \ell_2' \in Loc_2$, $s \in [\![V^L]\!]$, $g \in [\![V^G]\!]$ and $a \in \Sigma$:

$$\frac{(\ell_1, s) \xrightarrow{g\,a}_{\diamond,1} (\ell_1', \{s'\}) \qquad (\ell_2, s) \xrightarrow{g\,a}_{\diamond,2} (\ell_2', \{s'\})}{((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\diamond} ((\ell_1', \ell_2'), \{s'\})}[\mathsf{may}_\wedge]$$

$$\frac{(\ell_1, s) \xrightarrow{g\,a}_{\Box,1} (\ell_1', S_1') \qquad S' = \{s' \in S_1' \mid (\ell_2, s) \xrightarrow{g\,a}_{\diamond,2} (\ell_2', \{s'\})\} \neq \emptyset}{((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\Box} ((\ell_1', \ell_2'), S')}[\mathsf{must1}_\wedge]$$

$$\frac{(\ell_2, s) \xrightarrow{g\,a}_{\Box,2} (\ell_2', S_2') \qquad S' = \{s' \in S_2' \mid (\ell_1, s) \xrightarrow{g\,a}_{\diamond,1} (\ell_1', \{s'\})\} \neq \emptyset}{((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\Box} ((\ell_1', \ell_2'), S')}[\mathsf{must2}_\wedge]$$

$$\frac{(\ell_1, s) \xrightarrow{g\,a}_{\Box,1} (\ell_1', S_1') \qquad S' = \{s' \in S_1' \mid (\ell_2, s) \xrightarrow{g\,a}_{\diamond,2} (\ell_2', \{s'\})\} = \emptyset}{((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\Box} ((\ell_1', \ell_2), S')}[\mathsf{error1}_\wedge]$$

$$\frac{(\ell_2, s) \xrightarrow{g\,a}_{\Box,2} (\ell_2', S_2') \qquad S' = \{s' \in S_2' \mid (\ell_1, s) \xrightarrow{g\,a}_{\diamond,1} (\ell_1', \{s'\})\} = \emptyset}{((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\Box} ((\ell_1, \ell_2'), S')}[\mathsf{error2}_\wedge]$$

The $[\mathsf{error}\mathsf{X}_\wedge]$ rules are needed to capture exactly those data states where one component prevents the other from taking a given transition. This will give a must transition leading to a location with an empty data state.

The following theorem characterizes the relationship between the syntactic and semantic conjunction, under the assumption of determinism:

Theorem 6. *Let* $\mathbf{A}_1$, $\mathbf{A}_2$ *be two deterministic specifications with the same signature. Then* $\langle \mathbf{A}_1 \wedge \mathbf{A}_2 \rangle_{\mathrm{sem}} = \langle \mathbf{A}_1 \rangle_{\mathrm{sem}} \wedge_{\mathrm{sem}} \langle \mathbf{A}_2 \rangle_{\mathrm{sem}}$.

The proof for Thm. 6 can be found in Appendix A. Conjunction has the expected and desired properties of being both commutative and associative.

Refinement is a precongruence with respect to conjunction for deterministic specifications. Moreover, under the assumption of determinism, the conjunction construction yields the greatest lower bound with respect to modal refinement:

Theorem 7. *Let* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ *be specifications with the same signature and let* $\mathbf{A}$ *and* $\mathbf{B}$ *be deterministic. If* $\mathbf{A} \wedge \mathbf{B}$ *is consistent then*

1. $\rho(\mathbf{A} \wedge \mathbf{B}) \leq \mathbf{A}$ *and* $\rho(\mathbf{A} \wedge \mathbf{B}) \leq \mathbf{B}$,
2. $\mathbf{C} \leq \mathbf{A}$ *and* $\mathbf{C} \leq \mathbf{B}$ *implies* $\mathbf{C} \leq \rho(\mathbf{A} \wedge \mathbf{B})$,
3. $Impl(\rho(\mathbf{A} \wedge \mathbf{B})) = Impl(\mathbf{A}) \cap Impl(\mathbf{B})$.

Proof of Thm. 7. By Thm. 5 and Thm. 6 it suffices to consider the semantics of specifications, so let $\mathbf{S} = \langle \mathbf{A} \rangle_{\mathrm{sem}}$, $\mathbf{T} = \langle \mathbf{B} \rangle_{\mathrm{sem}}$ and $\mathbf{U} = \langle \mathbf{C} \rangle_{\mathrm{sem}}$.

1. We show $\rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T}) \leq_{\text{sem}} \mathbf{S}$, the other assertion is symmetric. We define a refinement relation $R \subseteq (Loc_{\mathbf{S}} \times Loc_{\mathbf{T}}) \times Loc_{\mathbf{S}} \times [\![V^L]\!]$ as follows:

$$R = \{((\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), \dot{\ell}_{\mathbf{S}}, s) \mid \ell_{\mathbf{S}} = \dot{\ell}_{\mathbf{S}}\}$$

Obviously, it holds that $S^0_{\rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})} \subseteq S^0_{\mathbf{S}}$, as $S^0_{\rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})} \subseteq S^0_{\mathbf{S}} \cap S^0_{\mathbf{T}}$.

Now, let $((\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), \ell_{\mathbf{S}}, s) \in R$. If, on the one hand,

$$((\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s) \xrightarrow{g\,a}_{\diamond, \rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})} ((\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}), \{s'\})$$

then this must come from the rule $[\mathsf{may}_\wedge]$, thus $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\diamond, \mathbf{S}} (\ell'_{\mathbf{S}}, \{s'\})$, and clearly $((\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}), \ell'_{\mathbf{S}}, s') \in R$. If, on the other hand, $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\Box, \mathbf{S}} (\ell'_{\mathbf{S}}, S')$, then we can apply the rule $[\mathsf{must1}_\wedge]$; if the rule $[\mathsf{error1}_\wedge]$ would be applicable then this would lead to contradiction with the assumption that $\rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})$ is syntactically consistent. Thus, by rule $[\mathsf{must1}_\wedge]$, we get

$$((\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s) \xrightarrow{g\,a}_{\Box,} ((\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}), S'')$$

and by the definition of S'' it clearly holds that $S'' \subseteq S'$, and $((\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}), \ell'_{\mathbf{S}}, s') \in R$ for all $s' \in S''$.

2. We can assume $\mathbf{U} \leq_{\text{sem}} \mathbf{S}$ and $\mathbf{U} \leq_{\text{sem}} \mathbf{T}$, and we show $\mathbf{U} \leq_{\text{sem}} \rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})$. We can assume refinement relations R_1 for $\mathbf{U} \leq_{\text{sem}} \mathbf{S}$ and R_2 for $\mathbf{U} \leq_{\text{sem}} \mathbf{T}$, then we define a relation $R \subseteq Loc_{\mathbf{C}} \times (Loc_{\mathbf{S}} \times Loc_{\mathbf{T}}) \times [\![V^L]\!]$ by

$$R = \{(\ell_{\mathbf{U}}, (\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s) \mid (\ell_{\mathbf{U}}, \ell_{\mathbf{S}}, s) \in R_1 \text{ and } (\ell_{\mathbf{U}}, \ell_{\mathbf{T}}, s) \in R_2\}.$$

We show that R witnesses $\mathbf{U} \leq_{\text{sem}} \rho_{\text{sem}}(\mathbf{S} \wedge \mathbf{T})$. Clearly, $(\ell_{\mathbf{U}}, (\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s) \in R$ for any $s \in S^0_{\mathbf{S}} \cap S^0_{\mathbf{T}}$. Now, let $(\ell_{\mathbf{U}}, (\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s) \in R$. It can be easily proven that every may-transition in $\mathbf{U}$ is simulated in $\mathbf{S} \wedge_{\text{sem}} \mathbf{T}$. The more interesting case is the other case: assume $((\ell_{\mathbf{S}}, \ell_{\mathbf{T}}), s) \xrightarrow{g\,a}_{\Box, \rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})} ((\ell'_S, \ell'_{\mathbf{T}}), S')$. Then this transition must come from (w.l.o.g.) $[\mathsf{must1}_\wedge]$, hence $(\ell_{\mathbf{S}}, s) \xrightarrow{g\,a}_{\Box, \mathbf{S}} (\ell'_{\mathbf{S}}, S'')$ and $S' = \{s' \in S'' \mid (\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\diamond, \mathbf{T}} (\ell'_{\mathbf{T}}, \{s'\})\}$. This transition must be simulated in $\mathbf{U}$, so it follows that $(\ell_{\mathbf{U}}, s) \xrightarrow{g\,a}_{\Box, \mathbf{U}} (\ell'_C, C')$ such that $C' \subseteq S''$ and $(\ell'_{\mathbf{U}}, \ell'_{\mathbf{S}}, s') \in R_1$ for every $s' \in C'$. It remains to show that $C' \subseteq S'$: Assume that there exists $\dot{s} \in C' \setminus S'$, then there must be a may-transition $(\ell_{\mathbf{U}}, s) \xrightarrow{g\,a}_{\diamond, \mathbf{U}} (\ell'_{\mathbf{U}}, \{\dot{s}\})$ which must be simulated in $\mathbf{T}$ implying that there exists $(\ell_{\mathbf{T}}, s) \xrightarrow{g\,a}_{\diamond, \mathbf{T}} (\ell''_{\mathbf{T}}, \{\dot{s}\})$. But $\mathbf{T}$ is deterministic, hence $\ell''_{\mathbf{T}} = \ell'_{\mathbf{T}}$. We know $s' \in C' \subseteq S''$, then it follows that $s' \in S'$, contradiction. Finally, it is easy to see that $(\ell'_{\mathbf{U}}, (\ell'_{\mathbf{S}}, \ell'_{\mathbf{T}}), s') \in R$ for every $s' \in C'$.

3. $Impl(\rho_{\text{sem}}(\mathbf{S} \wedge_{\text{sem}} \mathbf{T})) = Impl(\mathbf{S}) \cap Impl(\mathbf{T})$ follows from the first and second assertion of this theorem.

□

Quotient as the dual operator to structural composition. The quotient operator allows factoring out behaviors from larger specifications. Given two specifications $\mathbf{A}$ and $\mathbf{B}$ the quotient of $\mathbf{B}$ by $\mathbf{A}$, in the following denoted $\mathbf{B} \setminus\!\setminus \mathbf{A}$, is the most general specification that can be composed with $\mathbf{A}$ such that the composition refines $\mathbf{B}$.

In the following, we assume for the signatures $Sig_{\mathbf{A}} = (\Sigma, V^L_{\mathbf{A}}, V^G_{\mathbf{A}})$ and $Sig_{\mathbf{B}} = (\Sigma, V^L_{\mathbf{B}}, V^G_{\mathbf{B}})$ that $V^L_{\mathbf{A}} \subseteq V^L_{\mathbf{B}}$. The signature of the quotient $\mathbf{B} \setminus\!\setminus \mathbf{A}$ is then $Sig_{\mathbf{B}\setminus\!\setminus\mathbf{A}} = (\Sigma, V^L_{\mathbf{B}\setminus\!\setminus\mathbf{A}}, V^G_{\mathbf{B}\setminus\!\setminus\mathbf{A}})$ with $V^L_{\mathbf{B}\setminus\!\setminus\mathbf{A}} = V^L_{\mathbf{B}} \setminus V^L_{\mathbf{A}}$ and $V^G_{\mathbf{B}\setminus\!\setminus\mathbf{A}} = V^G_{\mathbf{B}} \uplus V^L_{\mathbf{A}}$. Note that, as said before, we restrict ourselves to the case where $V^L_{\mathbf{A}} \uplus V^G_{\mathbf{A}} = V^L_{\mathbf{B}} \uplus V^G_{\mathbf{B}}$.

In our general model of specifications it is unknown whether a finite quotient exists. For specifications involving variables with finite domains only we define a semantic quotient operation which works on the (finite) semantics of $\mathbf{A}$ and $\mathbf{B}$. As already noticed in previous works, e.g. [21], non-determinism is problematic for quotienting, and thus specifications are assumed to be deterministic. In our case, even when assuming deterministic specifications, the non-determinism with respect to the next local data state is still there: thus the quotient $\mathbf{B} \setminus\!\setminus \mathbf{A}$, when performing a transition, does not know the next data state of $\mathbf{A}$. However, due to our semantics, in which transitions are guarded by uncontrolled states, the quotient can always observe the current data state of $\mathbf{A}$. This extension of the usual quotient can be shown to satisfy the following soundness and maximality property: Given MSDs $\mathbf{S}$ and $\mathbf{T}$ such that $\mathbf{S}$ is deterministic and $\mathbf{T} \setminus\!\setminus_{\text{sem}} \mathbf{S}$ is consistent, and the semantic pruning operator ρ_{sem}. Then $\mathbf{X} \leq_{\text{sem}} \rho_{\text{sem}}(\mathbf{T} \setminus\!\setminus_{\text{sem}} \mathbf{S})$ if and only if $\mathbf{S} \parallel_{\text{sem}} \mathbf{X} \leq_{\text{sem}} \mathbf{T}$ for any MSD $\mathbf{X}$.

The semantic quotienting operator is defined as follows:

Let V_1, V_2 be two sets of variables such that $V_1 \subseteq V_2$. For sets $S_1 \subseteq [\![V_1]\!], S_2 \subseteq [\![V_2]\!]$ we use the notation $S_2 \setminus\!\setminus S_1$ for the set $\{s \in [\![V_2 \setminus V_1]\!] \mid \forall s_1 \in S_1 : (s_1 \cdot s) \in S_2\}$. It is easy to see that for any $S \subseteq [\![V_2 \setminus V_1]\!]$, it is satisfied that $(S_1 \cdot S) \subseteq S_2$ if and only if $S \subseteq S_2 \setminus\!\setminus S_1$.

Let $\mathbf{T}$ and $\mathbf{S}$ be two MSDs such that $V^L_{\mathbf{S}} \subseteq V^L_{\mathbf{T}}$. The *quotient of* $\mathbf{T}$ *by* $\mathbf{S}$ is defined as the possibly syntactically inconsistent MSD $\mathbf{T} \setminus\!\setminus_{\text{sem}} \mathbf{S} = (Sig, (Loc_{\mathbf{T}} \times Loc_{\mathbf{S}} \times \mathscr{P}_{\geq 1}([\![V^L_{\mathbf{S}}]\!])) \cup \{\mathsf{univ}, \bot\}, (\ell^0_{\mathbf{T}}, \ell^0_{\mathbf{S}}, S^0_{\mathbf{S}}), S^0_{\mathbf{T}} \setminus\!\setminus S^0_{\mathbf{S}}, \longrightarrow_{\diamond}, \longrightarrow_{\Box})$ where Sig is like in Def. 7, univ is a new *universal* state, $\bot$ is a new *error* state, and the transition relations are the smallest relations satisfying, for all $(\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S) \in (Loc_{\mathbf{T}} \times Loc_{\mathbf{S}} \times \mathscr{P}_{\geq 1}([\![V^L_{\mathbf{S}}]\!]))$ and all $q \in [\![V^L_{\mathbf{T}\setminus\!\setminus_{\text{sem}}\mathbf{S}}]\!]$:

$$\frac{(\ell_{\mathbf{T}}, s \cdot q) \xrightarrow{g\ a}_{\diamond,\mathbf{T}} (\ell'_{\mathbf{T}}, T') \quad (\ell_{\mathbf{S}}, s) \xrightarrow{(q\cdot g)\ a}_{\diamond,\mathbf{S}} (\ell'_{\mathbf{S}}, S') \quad s \in S \subseteq [\![V^L_{\mathbf{S}}]\!], T' \setminus\!\setminus S' \neq \emptyset}{((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S), q) \xrightarrow{(g\cdot s)\ a}_{\diamond} ((\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S'), T' \setminus\!\setminus S')} [\mathsf{may}_{\setminus\!\setminus}]$$

$$\frac{(\ell_{\mathbf{T}}, s \cdot q) \xrightarrow{g\ a}_{\Box,\mathbf{T}} (\ell'_{\mathbf{T}}, T') \quad (\ell_{\mathbf{S}}, s) \xrightarrow{(q\cdot g)\ a}_{\Box,\mathbf{S}} (\ell'_{\mathbf{S}}, S') \quad s \in S \subseteq [\![V^L_{\mathbf{S}}]\!], T' \setminus\!\setminus S' \neq \emptyset}{((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S), q) \xrightarrow{(g\cdot s)\ a}_{\Box} ((\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S'), T' \setminus\!\setminus S')} [\mathsf{must}_{\setminus\!\setminus}]$$

$$\frac{(\ell_{\mathbf{T}}, s \cdot q) \xrightarrow{g\ a}_{\Box,\mathbf{T}} (\ell'_{\mathbf{T}}, T') \quad (\ell_{\mathbf{S}}, s) \xrightarrow{(q\cdot g)\ a}_{\Box,\mathbf{S}} (\ell'_{\mathbf{S}}, S') \quad s \in S \subseteq [\![V^L_{\mathbf{S}}]\!], T' \setminus\!\setminus S' = \emptyset}{((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S), q) \xrightarrow{(g\cdot s)\ a}_{\Box} (\bot, \{q\})} [\mathsf{error1}_{\setminus\!\setminus}]$$

$$\frac{(\ell_{\mathbf{T}}, s \cdot q) \xrightarrow{g\ a}_{\Box,\mathbf{T}} \quad (\ell_{\mathbf{S}}, s) \not\xrightarrow{(q\cdot g)\ a}_{\Box,\mathbf{S}} \quad s \in S \subseteq [\![V^L_{\mathbf{S}}]\!]}{((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S), q) \xrightarrow{(g\cdot s)\ a}_{\Box} (\bot, \{q\})} [\mathsf{error2}_{\setminus\!\setminus}]$$

$$\frac{s \notin S \subseteq [\![V^L_{\mathbf{S}}]\!] \qquad q' \in [\![V^L_{\mathbf{T}\setminus\!\setminus_{\text{sem}}\mathbf{S}}]\!]}{((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S), q) \xrightarrow{(g\cdot s)\ a}_{\diamond} (\mathsf{univ}, \{q'\})} [\mathsf{data\text{-}unreachable}_{\setminus\!\setminus}]$$

$$\frac{(\ell_{\mathbf{S}}, s) \not\xrightarrow{(q\cdot g)\,a}_{\Diamond,\mathbf{S}} \qquad s \in S \subseteq [\![V^L_{\mathbf{S}}]\!], q' \in [\![V^L_{\mathbf{T}\backslash\!\backslash_{\mathrm{sem}}\mathbf{S}}]\!]}{((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S), q) \xrightarrow{(g\cdot s)\,a}_{\Diamond} (\mathsf{univ}, \{q'\})}[\mathsf{unreachable}_{\backslash\!\backslash}]$$

$$\frac{g \in [\![V^G]\!], q' \in [\![V^L_{\mathbf{T}\backslash\!\backslash_{\mathrm{sem}}\mathbf{S}}]\!]}{(\mathsf{univ}, q) \xrightarrow{g\,a}_{\Diamond} (\mathsf{univ}, \{q'\})}[\mathsf{universal}_{\backslash\!\backslash}]$$

$$\frac{g \in [\![V^G]\!]}{(\bot, q) \xrightarrow{g\,a}_{\Box} (\bot, [\![V^L_{\mathbf{T}\backslash\!\backslash_{\mathrm{sem}}\mathbf{S}}]\!])}[\mathsf{errorstate}_{\backslash\!\backslash}]$$

Theorem 8. *Let* $\mathbf{X}, \mathbf{S}, \mathbf{T}$ *be MSDs such that* $\mathbf{S}$ *is deterministic and* $\mathbf{T} \mathbin{\backslash\!\backslash_{\mathrm{sem}}} \mathbf{S}$ *is consistent. Then* $\mathbf{X} \leq_{\mathrm{sem}} \rho(\mathbf{T} \mathbin{\backslash\!\backslash_{\mathrm{sem}}} \mathbf{S})$ *if and only if* $\mathbf{S} \parallel_{\mathrm{sem}} \mathbf{X} \leq_{\mathrm{sem}} \mathbf{T}$.

In the following theorem we use the quotient of two specifications as given in Definition 7.

Theorem 9. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be specifications such that* $V^L_{\mathbf{A}} \subseteq V^L_{\mathbf{B}}$ *and such that all transition predicates in* $\mathbf{A}$ *and* $\mathbf{B}$ *are separable. Then the reachable part of* $\langle \mathbf{B} \mathbin{\backslash\!\backslash} \mathbf{A} \rangle_{\mathrm{sem}}$ *equals* $\langle \mathbf{B} \rangle_{\mathrm{sem}} \mathbin{\backslash\!\backslash_{\mathrm{sem}}} \langle \mathbf{A} \rangle_{\mathrm{sem}}$ *up to isomorphism.*

The proofs for Theorems 8 and 9 can be found in Appendix A.

Now our goal is to compute the quotient at the symbolic level of specifications. We do this for a restricted subclass of specifications in which each occurring transition predicate ψ is *separable*, meaning that ψ is equivalent to $^\circ\psi \wedge \psi^\circ$. Although this might seem as a serious restriction, we can often transform transition systems with transition predicates of the form $(x)' = x + 1$ to transition systems with separable transition predicates while keeping the same set of implementations. For instance, if we know that there are only finitely many possible values $v_1, \ldots, v_n$ for x in the current state, we can "unfold" the specification and replace the transition predicate $(x)' = x + 1$ by $(x)' = v_i + 1$, for $1 \leq i \leq n$.Thus we would get n transitions, but all of them with separable transition predicates.

The symbolic quotient introduces two new locations, the universal state (univ) and an error state ($\bot$). In the universal state the quotient can show arbitrary behavior and is needed to obtain maximality, and the error state is a syntactically inconsistent state used to encode conflicting requirements. The state space of the quotient is given by $\mathit{Loc}_{\mathbf{B}} \times \mathit{Loc}_{\mathbf{A}} \times \mathit{Pred}(V^L_{\mathbf{A}})$, so every state stores not only the current location of $\mathbf{B}$ and $\mathbf{A}$ (like in [21]) but includes a predicate about the current possible data states of $\mathbf{A}$. For notational convenience, for $\varphi \in \mathit{Pred}(V_1 \uplus V_2)$ and $\varphi_1 \in \mathit{Pred}(V_1)$, we write $\varphi \mathbin{\backslash\!\backslash} \varphi_1$ for $(\forall V_1.\varphi_1 \Rightarrow \varphi) \in \mathit{Pred}(V_2)$.

Definition 7. Let $\mathbf{A}$ and $\mathbf{B}$ be two specifications such that $V^L_{\mathbf{A}} \subseteq V^L_{\mathbf{B}}$. The *quotient of* $\mathbf{B}$ *by* $\mathbf{A}$ is defined as the possibly syntactically inconsistent specification $\mathbf{B} \mathbin{\backslash\!\backslash} \mathbf{A} = (\mathit{Sig}_{\mathbf{B}\backslash\!\backslash\mathbf{A}}, (\mathit{Loc}_{\mathbf{B}} \times \mathit{Loc}_{\mathbf{A}} \times \mathit{Pred}(V^L_{\mathbf{A}})) \cup \{\mathsf{univ}, \bot\}, (\ell^0_{\mathbf{B}}, \ell^0_{\mathbf{A}}, \varphi^0_{\mathbf{A}}), \varphi^0_{\mathbf{B}} \mathbin{\backslash\!\backslash} \varphi^0_{\mathbf{A}}, E_\Diamond, E_\Box)$ where the transition relations are given by, for all $a \in \Sigma$ and all $\xi_{\mathbf{A}} \in \mathit{Pred}(V^L_{\mathbf{A}})$,

1. if $(\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Diamond,\mathbf{B}}$ and $(\ell_{\mathbf{A}}, a, \psi_{\mathbf{A}}, \ell'_{\mathbf{A}}) \in E_{\Diamond,\mathbf{A}}$, then $((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \xi_{\mathbf{A}} \wedge {}^\circ\psi_{\mathbf{B}} \wedge {}^\circ\psi_{\mathbf{A}} \wedge (\psi^\circ_{\mathbf{B}} \mathbin{\backslash\!\backslash} \psi^\circ_{\mathbf{A}}), (\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, \psi^\circ_{\mathbf{A}}\!\downarrow)) \in E_\Diamond$,
2. if $(\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Box,\mathbf{B}}$ and $(\ell_{\mathbf{A}}, a, \psi_{\mathbf{A}}, \ell'_{\mathbf{A}}) \in E_{\Box,\mathbf{A}}$, then $((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \xi_{\mathbf{A}} \wedge {}^\circ\psi_{\mathbf{B}} \wedge {}^\circ\psi_{\mathbf{A}} \wedge (\psi^\circ_{\mathbf{B}} \mathbin{\backslash\!\backslash} \psi^\circ_{\mathbf{A}}), (\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, \psi^\circ_{\mathbf{A}}\!\downarrow)) \in E_\Box$,

3. if $(\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Box,\mathbf{B}}$ and $(\ell_{\mathbf{A}}, a, \psi_{\mathbf{A}}, \ell'_{\mathbf{A}}) \in E_{\Box,\mathbf{A}}$, then $((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \xi_{\mathbf{A}} \wedge {}^\circ\psi_{\mathbf{B}} \wedge {}^\circ\psi_{\mathbf{A}} \wedge \neg(\psi^\circ_{\mathbf{B}} \setminus\!\!\setminus \psi^\circ_{\mathbf{A}}), \bot) \in E_\Box$,

4. if $(\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Box,\mathbf{B}}$, then $((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \xi_{\mathbf{A}} \wedge {}^\circ\psi_{\mathbf{B}} \wedge \bigwedge_{\psi_{\mathbf{A}} \in M} \neg^\circ\psi_{\mathbf{A}}, \bot) \in E_\Box$ where $M = \bigcup_{\ell'_{\mathbf{A}} \in Loc_{\mathbf{A}}} Must^a_{\mathbf{A}}(\ell_{\mathbf{A}}, \ell'_{\mathbf{A}})$,

5. $((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \neg\xi_{\mathbf{A}}, \mathsf{univ}) \in E_\Diamond$,

6. $((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \xi_{\mathbf{A}} \wedge \bigwedge_{\psi_{\mathbf{A}} \in M} \neg^\circ\psi_{\mathbf{A}}, \mathsf{univ}) \in E_\Diamond$ where $M = \bigcup_{\ell'_{\mathbf{A}} \in Loc_{\mathbf{A}}} May^a_{\mathbf{A}}(\ell_{\mathbf{A}}, \ell'_{\mathbf{A}})$,

7. $(\mathsf{univ}, a, true, \mathsf{univ}) \in E_\Diamond$,

8. $(\bot, a, true, \bot) \in E_\Box$.

Rules (1) and (2) capture the cases when both $\mathbf{A}$ and $\mathbf{B}$ can perform a may- and must-transition, respectively. Rules (3) and (4) capture any inconsistencies which can arise if for a must-transition in $\mathbf{B}$ there is no way to obtain a must-transition by composition of the quotient with $\mathbf{A}$. In order to obtain maximality, we add a universal state univ in which the behavior of the quotient is not restricted (rules (5)–(7)). Finally, the rule (8) makes the error state syntactically inconsistent.

Since we only have finitely many transition predicates $\psi_{\mathbf{A}}$ in $\mathbf{A}$, and they are all separable, the set of locations $(Loc_{\mathbf{B}} \times Loc_{\mathbf{A}} \times (\{\psi^\circ_{\mathbf{A}}\downarrow \mid \psi_{\mathbf{A}} \text{ occurring in } \mathbf{A}\} \cup \{\varphi^0_{\mathbf{A}}\})) \cup \{\mathsf{univ}, \bot\}$ of $\mathbf{B} \setminus\!\!\setminus \mathbf{A}$ is also finite. Thus we can construct the symbolic quotient in a finite number of steps, starting in the initial state $(\ell^0_{\mathbf{B}}, \ell^0_{\mathbf{A}}, \varphi^0_{\mathbf{A}})$, and iteratively constructing the transitions. Soundness and maximality of the quotient follows from the following theorem.

Theorem 10. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be specifications such that* $V^L_{\mathbf{A}} \subseteq V^L_{\mathbf{B}}$*, all transition predicates of* $\mathbf{A}$ *and* $\mathbf{B}$ *are separable,* $\mathbf{A}$ *is deterministic and* $\mathbf{B} \setminus\!\!\setminus \mathbf{A}$ *is consistent. Then for any specification* $\mathbf{C}$ *such that* $Sig_{\mathbf{C}} = Sig_{\mathbf{B} \setminus\!\!\setminus \mathbf{A}}$*,* $\mathbf{C} \leq \rho(\mathbf{B} \setminus\!\!\setminus \mathbf{A})$ *if and only if* $\mathbf{A} \parallel \mathbf{C} \leq \mathbf{B}$.

Proof of Thm. 10. This follows from Theorems 2, 5, 8 and 9:
$\mathbf{C} \leq \rho(\mathbf{B} \setminus\!\!\setminus \mathbf{A})$
iff $\langle\mathbf{C}\rangle_{\text{sem}} \leq_{\text{sem}} \langle\rho(\mathbf{B} \setminus\!\!\setminus \mathbf{A})\rangle_{\text{sem}}$
iff $\langle\mathbf{C}\rangle_{\text{sem}} \leq_{\text{sem}} \rho_{\text{sem}}(\langle\mathbf{B}\rangle_{\text{sem}} \setminus\!\!\setminus_{\text{sem}} \langle\mathbf{A}\rangle_{\text{sem}})$
iff $\langle\mathbf{A}\rangle_{\text{sem}} \parallel_{\text{sem}} \langle\mathbf{C}\rangle_{\text{sem}} \leq_{\text{sem}} \langle\mathbf{B}\rangle_{\text{sem}}$
iff $\mathbf{A} \parallel \mathbf{C} \leq \mathbf{B}$. □

4. Predicate Abstraction for Verification of Refinement

We now switch our focus to the problem of deciding whether a specification $\mathbf{A}$ refines another specification $\mathbf{B}$ (which reduces to checking $\langle\mathbf{A}\rangle_{\text{sem}} \leq_{\text{sem}} \langle\mathbf{B}\rangle_{\text{sem}}$). As soon as domains of variables are infinite, $\langle\mathbf{A}\rangle_{\text{sem}}$ and $\langle\mathbf{B}\rangle_{\text{sem}}$ may be MSDs with infinitely many states and transitions. In this case, this problem is known to be undecidable in general. Thus we propose to resort to predicate abstraction techniques [22].

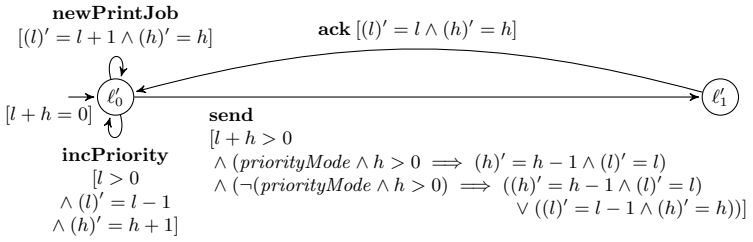


Figure 7: Refined print server specification $\mathbf{Q}$.

Given two specifications $\mathbf{A}$ and $\mathbf{B}$ we derive over- and under-approximations $\mathbf{A}^o$ and $\mathbf{B}^u$ which are guaranteed to be *finite* MSDs. Then, we show that $\mathbf{A}^o \leq_{\text{sem}} \mathbf{B}^u$ implies $\mathbf{A} \leq \mathbf{B}$.

Example 5. Fig. 7 shows a print server specification $\mathbf{Q}$ which we will show is a refinement of the abstract specification $\mathbf{P}$ in Fig. 3. The behavior of the print server is now fixed for any number of print jobs. Moreover, the send transition has been refined such that depending on the priority mode (provided by the environment of the print server) a job with high priority (in case $priorityMode$ is true) or a job with low priority (otherwise) is chosen next.

Given a specification $\mathbf{A} = (Sig, Loc, \ell^0, \varphi^0, \longrightarrow_\diamond, \longrightarrow_\Box)$ with $Sig = (\Sigma, V^L, V^G)$, we partition the local state space and the uncontrolled state space using finitely many predicates $\phi_1, \phi_2, \ldots, \phi_N \in Pred(V^L)$ and $\chi_1, \chi_2, \ldots, \chi_M \in Pred(V^G)$. We fix these predicates in the following to simplify the presentation. The signature of the abstraction is then given by $Sig_{abstr} = (\Sigma, V^L_{abstr}, V^G_{abstr})$, where $V^L_{abstr} = \{x_1, x_2, \ldots, x_N\}$ and $V^G_{abstr} = \{y_1, y_2, \ldots, y_M\}$. All variables x_i, y_j have Boolean domain. A variable x_i (y_j) encodes whether the predicate ϕ_i (χ_j) holds or not.

Any abstract state $\nu \in [\![V^L_{abstr}]\!]$ is a conjunction of predicates $\bigwedge_{i=1}^N \phi_i^{\nu(x_i)}$, where $\phi_i^{\nu(x_i)} = \phi_i$ if $\nu(x_i) = 1$, else $\phi_i^{\nu(x_i)} = \neg\phi_i$. Further, a set of abstract states $N \subseteq [\![V^L_{abstr}]\!]$ corresponds to $\bigvee_{\nu \in N} \nu$. Similarly for any $\omega \in [\![V^G_{abstr}]\!]$ and for $M \subseteq [\![V^G_{abstr}]\!]$.

The transition relation of the over-approximation expands the allowed behaviors and limits the required behaviors. Dually, the under-approximation will further restrict the allowed behavior and add more required transitions. In other words, over-approximation is an *existential* abstraction on may-transitions and *universal* abstraction on must-transitions; dually for the under-approximation. Table 1 illustrates both abstractions.

Formally, the *over-approximation* $\mathbf{A}^o$ of $\mathbf{A}$ is defined by the finite MSD $(Sig_{abstr}, Loc, \ell^0, S^0_{abstr}, \longrightarrow_{\diamond,abstr}, \longrightarrow_{\Box,abstr})$, where the initial abstract state S^0_{abstr} contains all partitions containing some concrete initial state, i.e. the initial abstract state is defined by $S^0_{abstr} = \{\nu \in [\![V^L_{abstr}]\!] \mid \exists V^L.\nu \wedge \varphi^0\}$, and the abstract transition relations are derived as follows. For all $\ell, \ell' \in Loc$, $a \in Act$, $\nu, \dot{\nu} \in [\![V^L_{abstr}]\!]$, $\omega \in [\![V^G_{abstr}]\!]$,

i. If $\exists V.\exists (V^L)'.\nu \wedge \omega \wedge (\bigvee_{\psi \in May^a(\ell,\ell')} \psi) \wedge (\dot{\nu})'$, then $(\ell, \nu) \xrightarrow{\omega\, a}_{\diamond,abstr} (\ell', \{\dot{\nu}\})$, so there is a may-transition between partitions in the abstraction if there was a may-transition between any states in these partitions in the concrete system.

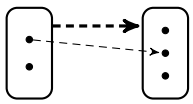
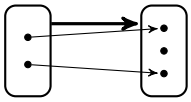
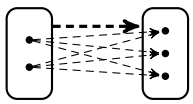
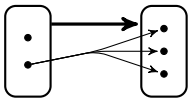
	may-transitions $\longrightarrow_{\Diamond}$	must-transitions $\longrightarrow_{\Box}$
Over-approximation		
Under-approximation		

Table 1: Over- and under-approximation schematically, • represents concrete states, ▢ represents abstract states. Upper left: If a single may-transition from a concrete state in one abstract state can reach a concrete state in another abstract state then the two abstract states are connected with a may transition in the over approximation. Lower left: Every concrete state in an abstract state must individually be able to reach every state in another abstract state before the two are related with a may-transition in the under-approximation. Upper right: Every concrete state in an abstract state must have a must-transition going to some state in another abstract state for them to be connected by a must-transition in the over-approximation. Lower right: At least one concrete state in an abstract state must have a must-transition that covers the entire target abstract state for them to be linked by must-transition in the under-approximation.

ii. Whenever, for some $N \subseteq [\![V^L_{abstr}]\!]$, the predicate

$$\forall V.\nu \wedge \omega \Rightarrow \bigvee_{\psi \in Must^a(\ell,\ell')} {}^\circ\psi \wedge (\forall (V^L)'.\psi \Rightarrow (N)')$$

is true and N is minimal with respect to this property, then $(\ell,\nu) \xrightarrow{\omega\, a}_{\Box,abstr} (\ell', N)$.

For the *under-approximation* $\mathbf{B}^u$ of $\mathbf{B}$, we assume that every transition predicate ψ on a must-transition must be separable (see page 21). Moreover, in order to soundly capture must-transitions, we must be able to exactly describe the target set of (concrete) local states by a union of abstract states; so for any $(\ell, a, \psi, \ell') \in E_{\Box,\mathbf{B}}$, there exists a set $N \subseteq [\![V^L_{abstr}]\!]$ such that $\forall (V^L)'.\, \psi^\circ \Leftrightarrow (N)'$. The under-approximation $\mathbf{B}^u$ is the finite MSD $(Sig_{abstr}, Loc, \ell^0, S^0_{abstr}, \longrightarrow_{\diamond,abstr}, \longrightarrow_{\Box,abstr})$, where $S^0_{abstr} = \{\nu \in [\![V^L_{abstr}]\!] \mid \forall V^L.\nu \Rightarrow \varphi^0\}$, and for all $\ell, \ell' \in Loc$, $a \in Act$, $\nu, \dot\nu \in [\![V^L_{abstr}]\!]$, $\omega \in [\![V^G_{abstr}]\!]$,

i. If $\forall V.\forall (V^L)'.\nu \wedge \omega \wedge (\dot\nu)' \Rightarrow \bigvee_{\psi \in May^a(\ell,\ell')} \psi$ then $(\ell,\nu) \xrightarrow{\omega\, a}_{\diamond,abstr} (\ell', \{\dot\nu\})$,

ii. For every $(\ell, a, \psi, \ell') \in E_{\Box,}$, if $\exists V.\nu \wedge \omega \wedge {}^\circ\psi$, then $(\ell,\nu) \xrightarrow{\omega\, a}_{\Box,abstr} (\ell', N)$ where $N \subseteq [\![V^L_{abstr}]\!]$ such that $\forall (V^L)'.\psi^\circ \Leftrightarrow (N)'$.

Correctness of the abstraction follows from the following theorem.

Theorem 11. $\mathbf{A}^o \leq_{\mathrm{sem}} \mathbf{B}^u$ *implies* $\mathbf{A} \leq \mathbf{B}$.

Proof of Thm. 11. Technically, under-approximation may yield a syntactically inconsistent MSD in which targets reachable by must-transitions are not reachable by may-transitions. However, this does not affect the following proof.

We can assume a relation R' witnessing $\mathbf{A}^o \leq \mathbf{B}^u$. We define a relation $R \subseteq Loc_{\mathbf{A}} \times Loc_{\mathbf{B}} \times [\![V^L]\!]$ by

$$R = \{(\ell_{\mathbf{A}}, \ell_{\mathbf{B}}, s) \mid (\ell_{\mathbf{A}}, \ell_{\mathbf{B}}, \nu) \in R' \text{ such that } \nu(s)\}$$

and we show that R witnesses $\mathbf{A} \leq \mathbf{B}$, more precisely, $\langle \mathbf{A} \rangle_{\mathrm{sem}} \leq \langle \mathbf{B} \rangle_{\mathrm{sem}}$.

First, $S^0_{\mathbf{A}} \subseteq S^0_{\mathbf{B}}$: assume $s \in S^0_{\mathbf{A}}$ then there exists $\nu \in [\![V^L_{abstr}]\!]$ such that s satisfies ν. Then, by R' we know that $\nu \in S^0_{\mathbf{A}^o} \subseteq S^0_{\mathbf{B}^u}$, so $\nu \in S^0_{\mathbf{B}^u}$. This implies that $s \in S^0_{\mathbf{B}}$. Moreover, $(\ell_{\mathbf{A}}, \ell_{\mathbf{B}}, s) \in R$ for any $s \in S^0_{\mathbf{A}}$.

Now, let $(\ell_{\mathbf{A}}, \ell_{\mathbf{B}}, s) \in R$. We can assume that $(\ell_{\mathbf{A}}, \ell_{\mathbf{B}}, \nu) \in R'$ for some ν for which $\nu(s)$.

1. This direction is straightforward and omitted here.

2. Assume

$$(\ell_{\mathbf{B}}, s) \xrightarrow{g\, a}_{\Box,\langle \mathbf{B} \rangle_{\mathrm{sem}}} (\ell'_{\mathbf{B}}, S').$$

Then there exists $(\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Box,\mathbf{B}}$ such that ${}^\circ\psi_{\mathbf{B}}(s \cdot g)$ and $S' = [\![\psi^\circ_{\mathbf{B}}]\!] \neq \emptyset$. Then there exists $(\ell_{\mathbf{B}}, \nu) \xrightarrow{\omega\, a}_{\Box,\mathbf{B}^u} (\ell'_{\mathbf{B}}, N)$ such that $\omega(g)$ and $\forall (V^L)'.\psi^\circ_{\mathbf{B}} \Leftrightarrow (N)'$. From $(\ell_{\mathbf{A}}, \ell_{\mathbf{B}}, \nu) \in R'$ we can conclude that there exists

$$(\ell_{\mathbf{A}}, \nu) \xrightarrow{\omega\, a}_{\Box,\mathbf{A}^o} (\ell'_{\mathbf{A}}, \dot N)$$

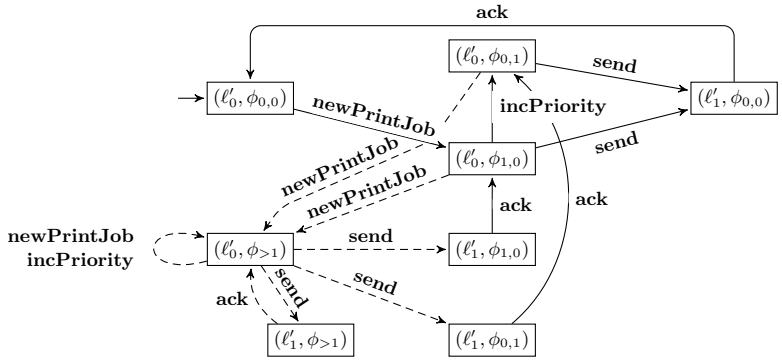


Figure 8: Over-approximation $\mathbf{Q}^o$.

such that $\dot{N} \subseteq N$ and $(\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}, \dot{\nu}) \in R'$ for any $\dot{\nu} \in \dot{N}$. By definition of the over-approximation, we get that $\dot{N}$ is a minimal set of abstract states, satisfying

$$\forall V.\nu \wedge \omega \Rightarrow \bigvee\nolimits_{\psi_{\mathbf{A}} \in Must^a_{\mathbf{A}}(\ell,\ell')} \Big((\exists (V^L)'.\psi_{\mathbf{A}}) \wedge (\forall (V^L)'.\psi_{\mathbf{A}} \Rightarrow (\dot{N})') \Big).$$

Since $(\nu \wedge \omega)(s \cdot g)$, there exists a must-transition $(\ell_{\mathbf{A}}, a, \psi_{\mathbf{A}}, \ell'_{\mathbf{A}}) \in E_{\Box,\mathbf{A}}$ such that ${}^\circ\psi_{\mathbf{A}}(s \cdot g)$ and for any $s' \in [\![V^L]\!]$, $\psi_{\mathbf{A}}(s \cdot g, s')$ implies $\dot{N}(s')$. Since $\psi_{\mathbf{A}} \equiv {}^\circ\psi_{\mathbf{A}} \wedge \psi^\circ_{\mathbf{A}}$, it follows that $\forall (V^L)'.\psi^\circ_{\mathbf{A}} \Rightarrow \dot{N}$. Then we get

$$(\ell_{\mathbf{A}}, s) \xrightarrow{g\,a}_{\Box,\langle\mathbf{A}\rangle_{\mathrm{sem}}} (\ell'_{\mathbf{A}}, [\![\psi^\circ_{\mathbf{A}}]\!])$$

and $[\![\psi^\circ_{\mathbf{A}}]\!] \subseteq [\![\dot{N}]\!] \subseteq [\![N]\!] = [\![\psi^\circ_{\mathbf{B}}]\!] = S'$, and finally $(\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}, s') \in R$ for all $s' \in [\![\psi^\circ_{\mathbf{A}}]\!]$.

□

Example 6. Fig. 8 and Fig. 9 are over- and under-approximations of $\mathbf{Q}$ and $\mathbf{P}$, respectively. The MSDs represent abstractions w.r.t. the predicates $\phi_{0,0} =_{def} h = l = 0$, $\phi_{0,1} =_{def} l = 0 \wedge h = 1$, $\phi_{1,0} =_{def} l = 1 \wedge h = 0$, and $\phi_{>1} =_{def} h + l > 1$ for the controlled variables l and h, and $\omega_1 =_{def} priorityMode$, $\omega_2 =_{def} \neg priorityMode$ for the uncontrolled variable $priorityMode$. Note that all transition predicates in $\mathbf{P}$ are separable, and all possible (concrete) poststates can be precisely captured by the predicates $\phi_{0,0}, \phi_{0,1}, \phi_{1,0}, \phi_{>1}$. For better readability we have omitted most of the guards ω_1, ω_2, i.e. every transition without guard stands for two transitions with the same action, source and target state(s), and with ω_1 and ω_2 as guard, respectively. Moreover, the state $(\ell_3, \phi_{0,0} \vee \phi_{0,1} \vee \phi_{1,0} \vee \phi_{>1})$ is a simplified notation which represents all the states (ℓ_3, ϕ) with $\phi \in \{\phi_{0,0}, \phi_{0,1}, \phi_{1,0}, \phi_{>1}\}$ and all may-transitions leading to it lead to each of the states, and the may-loop stands for all the transitions between each of the states. Obviously, $\mathbf{Q}^o \leq_{\mathrm{sem}} \mathbf{P}^u$, and from Thm. 11 it follows that $\mathbf{Q} \leq \mathbf{P}$.

Even though this abstraction technique requires separability of predicates, it is applicable to a larger set of specifications. Sometimes, as already described in the previous section, transitions with non-separable predicates can be replaced by finite sets of

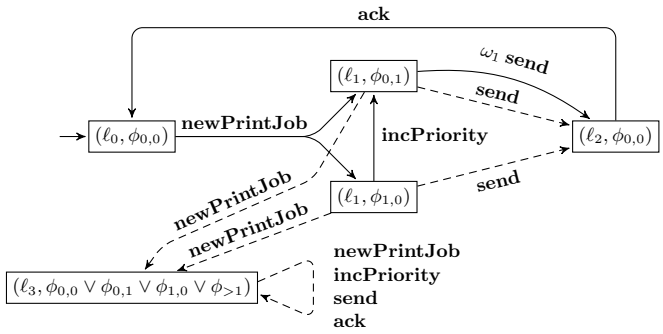


Figure 9: Under-approximation $\mathbf{P}^u$.

transitions to achieve separability, without changing the semantics of the specification. Automatic procedures for generation of predicates are subject of future work. Finally, our abstraction also supports compositional reasoning about parallel composition in the following sense:

Theorem 12. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two composable specifications, and* $V^G_{\mathbf{A}\|\mathbf{B}} = (V^G_{\mathbf{A}} \cup V^G_{\mathbf{B}}) \setminus (V^L_{\mathbf{A}} \uplus V^L_{\mathbf{B}})$*. Let* $E_{\mathbf{A}} \subseteq \mathit{Pred}(V^L_{\mathbf{A}})$*,* $E_{\mathbf{A}} \subseteq \mathit{Pred}(V^L_{\mathbf{B}})$*, and* $F \subseteq \mathit{Pred}(V^G_{\mathbf{A}\|\mathbf{B}})$ *be sets of predicates partitioning the respective data states.*

$\mathbf{A}$ *is approximated w.r.t.* $E_{\mathbf{A}}$ *for* $V^L_{\mathbf{A}}$*, and* $E_{\mathbf{B}} \cup F$ *for* $V^G_{\mathbf{A}} = V^G_{\mathbf{A}\|\mathbf{B}} \uplus V^L_{\mathbf{B}}$ *and similarly,* $\mathbf{B}$ *is approximated w.r.t.* $E_{\mathbf{B}}$ *and* $E_{\mathbf{A}} \cup F$*. Finally,* $\mathbf{A} \parallel \mathbf{B}$ *is approximated w.r.t.* $E_{\mathbf{A}} \cup E_{\mathbf{B}}$ *for* $V^L_{\mathbf{A}\|\mathbf{B}} = V^L_{\mathbf{A}} \uplus V^L_{\mathbf{B}}$*, and* F *for* $V^G_{\mathbf{A}\|\mathbf{B}}$*. We assume that each predicate, in any abstraction of* $\mathbf{A}$*,* $\mathbf{B}$*, or* $\mathbf{A} \parallel \mathbf{B}$*, is encoded with the same variable.*

Then $(\mathbf{A} \parallel \mathbf{B})^o \leq_{\mathrm{sem}} \mathbf{A}^o \parallel_{\mathrm{sem}} \mathbf{B}^o$*, and* $\mathbf{A}^u \parallel_{\mathrm{sem}} \mathbf{B}^u \leq_{\mathrm{sem}} (\mathbf{A} \parallel \mathbf{B})^u$.

This result allows reusing abstractions of individual components in a continued development and verification process. For instance, if we want to verify $\mathbf{A} \parallel \mathbf{B} \leq \mathbf{C}$ then we can compute (or reuse) the less complex abstractions $\mathbf{A}^o$ and $\mathbf{B}^o$. Thm. 12 implies then that from $\mathbf{A}^o \parallel_{\mathrm{sem}} \mathbf{B}^o \leq_{\mathrm{sem}} \mathbf{C}^u$ we can infer $\mathbf{A} \parallel \mathbf{B} \leq \mathbf{C}$.

Proof of Thm. 12. We prove $(\mathbf{A} \parallel \mathbf{B})^o \leq \mathbf{A}^o \parallel_{\mathrm{sem}} \mathbf{B}^o$: Let $V^L_{abstr(\mathbf{A})}$ the set of variables abstracting $V^L_{\mathbf{A}}$, $V^L_{abstr(\mathbf{B})}$ the set of variables for $V^L_{\mathbf{B}}$, and V^G_{abstr} the set of variables for $V^G_{\mathbf{A}} \setminus V^L_{\mathbf{B}}$. The witnessing relation $R \subseteq (\mathit{Loc}_{\mathbf{A}} \times \mathit{Loc}_{\mathbf{B}}) \times (\mathit{Loc}_{\mathbf{A}} \times \mathit{Loc}_{\mathbf{B}}) \times ([\![V^L_{abstr(\mathbf{A})}]\!] \times [\![V^L_{abstr(\mathbf{B})}]\!])$, defined by $R = \{((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), (\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}})\}$. First, $S^0_{(\mathbf{A}\|\mathbf{B})^o} \subseteq S^0_{\mathbf{A}^o \|_{\mathrm{sem}} \mathbf{B}^o}$: Let $\nu \in S^0_{(\mathbf{A}\|\mathbf{B})^o}$, then $\exists V^L_{abstr(\mathbf{A})}.\exists V^L_{abstr(\mathbf{B})}.\nu \wedge \varphi^0_{\mathbf{A}\|\mathbf{B}}$ implying $\exists V^L_{abstr(\mathbf{A})}.\nu_{\mathbf{A}} \wedge \varphi^0_{\mathbf{A}}$ and $\exists V^L_{abstr(\mathbf{B})}.\nu_{\mathbf{B}} \wedge \varphi^0_{\mathbf{B}}$ for $\nu = \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}}$. It follows that $\nu \in \varphi^0_{\mathbf{A}^o} \cdot \varphi^0_{\mathbf{B}^o}$.

Second, let $((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), (\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}}) \in R$.

1. The proof for the may-transition is straightforward and omitted.

2. Assume
$$((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}}) \xrightarrow{\omega\, a}_{\square, \mathbf{A}^o \parallel_{\mathrm{sem}} \mathbf{B}^o} ((\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), N).$$

Then there exist

$$(\ell_{\mathbf{A}}, \nu_{\mathbf{A}}) \xrightarrow{(\nu_{\mathbf{B}} \cdot \omega)\, a}_{\Box, \mathbf{A}^o} (\ell'_{\mathbf{A}}, N_{\mathbf{A}}) \text{ and } (\ell_{\mathbf{B}}, \nu_{\mathbf{B}}) \xrightarrow{(\nu_{\mathbf{A}} \cdot \omega)\, a}_{\Box, \mathbf{B}^o} (\ell'_{\mathbf{B}}, N_{\mathbf{B}})$$

such that $N = N_{\mathbf{A}} \cdot N_{\mathbf{B}}$. Then for every $s_{\mathbf{A}} \cdot s_{\mathbf{B}} \cdot g$ satisfying $\nu_{\mathbf{A}} \wedge \nu_{\mathbf{B}} \wedge \omega$ there exists $(\ell_{\mathbf{A}}, (a, \psi_{\mathbf{A}}), \ell'_{\mathbf{A}}) \in E_{\Box, \mathbf{A}}$ such that ${}^\circ\psi(s_{\mathbf{A}} \cdot s_{\mathbf{B}} \cdot g)$, and for all $s'_{\mathbf{A}} \in [\![V^L_{\mathbf{A}}]\!]$, $\psi(s_{\mathbf{A}} \cdot s_{\mathbf{B}} \cdot g, s'_{\mathbf{A}})$ implies $N_{\mathbf{A}}(s'_{\mathbf{A}})$. And similar we can conclude this for $\mathbf{B}$.

Then, by the definition of parallel composition, for every $s_{\mathbf{A}} \cdot s_{\mathbf{B}} \cdot g$ satisfying $\nu_{\mathbf{A}} \wedge \nu_{\mathbf{B}} \wedge \omega$ there exists

$$((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), (a, \psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}}), (\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}) \in E_{\Box, \mathbf{A} \| \mathbf{B}}$$

such that ${}^\circ\psi_{\mathbf{A}}(s_{\mathbf{A}} \cdot s_{\mathbf{B}} \cdot g)$ and for all $s_{\mathbf{A}} \cdot s_{\mathbf{B}}$, $(\psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}})(s_{\mathbf{A}} \cdot s_{\mathbf{B}} \cdot g, s'_{\mathbf{A}} \cdot s'_{\mathbf{B}})$ implies $(N_{\mathbf{A}} \cdot N_{\mathbf{B}})(s'_{\mathbf{A}} \cdot s'_{\mathbf{B}})$. Then, there exists also a minimal $N' \subseteq N_{\mathbf{A}} \cdot N_{\mathbf{B}} = N$ with this property, and then

$$((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}}) \xrightarrow{\omega\, a}_{\Box, (\mathbf{A} \parallel \mathbf{B})^o} ((\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), N'),$$

and $((\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), (\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), \nu'_{\mathbf{A}} \cdot \nu'_{\mathbf{B}}) \in R$ for all $\nu'_{\mathbf{A}} \cdot \nu'_{\mathbf{B}} \in N'$.

We prove now $\mathbf{A}^u \parallel_{\mathrm{sem}} \mathbf{B}^u \leq (\mathbf{A} \parallel \mathbf{B})^u$: The proof is in the same line as the previous proof, and for the refinement relation witnessing the claim, the same relation as before can be taken. We just check it for the must-transitions. Assume

$$((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}}) \xrightarrow{\omega\, a}_{\Box, (\mathbf{A} \parallel \mathbf{B})^u} ((\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), N).$$

Then there exists $((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), a, \psi, (\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}})) \in E_{\Box, \mathbf{A} \| \mathbf{B}}$ such that

$$\exists V. \nu_{\mathbf{A}} \wedge \nu_{\mathbf{B}} \wedge \omega \wedge {}^\circ\psi, \tag{2}$$

$$\forall (V^L_{\mathbf{A}})'. \forall (V^L_{\mathbf{B}})'. \psi^\circ \Leftrightarrow (N)'. \tag{3}$$

By the definition of parallel composition we get

$$(\ell_{\mathbf{A}}, a, \psi_{\mathbf{A}}, \ell'_{\mathbf{A}}) \in E_{\Box, \mathbf{A}} \text{ and } (\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Box, \mathbf{B}}$$

such that $\psi = \psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}}$. Then, from (2) and

$$\begin{aligned} {}^\circ(\psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}}) &= \exists (V^L_{\mathbf{A}})'. \exists (V^L_{\mathbf{B}})'. \psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}} \\ &= (\exists (V^L_{\mathbf{A}})'. \psi_{\mathbf{A}}) \wedge (\exists (V^L_{\mathbf{B}})'. \psi_{\mathbf{B}}) \\ &= {}^\circ\psi_{\mathbf{A}} \wedge {}^\circ\psi_{\mathbf{A}} \end{aligned}$$

it follows that

$$\exists V^L_{\mathbf{A}}. \exists V^G. \nu_{\mathbf{A}} \wedge \omega \wedge {}^\circ\psi_{\mathbf{A}} \text{ and } \exists V^L_{\mathbf{B}}. \exists V^G. \nu_{\mathbf{B}} \wedge \omega \wedge {}^\circ\psi_{\mathbf{B}}.$$

where $V^G = V^G_{\mathbf{A}} \setminus V^L_{\mathbf{B}}$. By our assumption that we can precisely capture the set of next local states of must-transition by a set of abstract states, we know that there

exists $\dot{N}_{\mathbf{A}} \subseteq [\![V^L_{abstr(\mathbf{A})}]\!]$ such that $\forall (V^L_{\mathbf{A}})'.(\dot{N}_{\mathbf{A}})' \Leftrightarrow \psi^\circ_{\mathbf{A}}$, and similarly, there exists $\dot{N}_{\mathbf{B}} \subseteq [\![V^L_{abstr(\mathbf{B})}]\!]$ such that $\forall (V^L_{\mathbf{B}})'.(\dot{N}_{\mathbf{B}})' \Leftrightarrow \psi^\circ_{\mathbf{B}}$. Then we have

$$(\ell_{\mathbf{A}}, \nu_{\mathbf{A}}) \xrightarrow{(\nu_{\mathbf{B}} \cdot \omega)\, a}_{\Box, \mathbf{A}^u} (\ell'_{\mathbf{A}}, \dot{N}_{\mathbf{A}}) \text{ and } (\ell_{\mathbf{B}}, \nu_{\mathbf{B}}) \xrightarrow{(\nu_{\mathbf{A}} \cdot \omega)\, a}_{\Box, \mathbf{B}^u} (\ell'_{\mathbf{B}}, \dot{N}_{\mathbf{B}}).$$

Then

$$((\ell_{\mathbf{A}}, \ell_{\mathbf{B}}), \nu_{\mathbf{A}} \cdot \nu_{\mathbf{B}}) \xrightarrow{\omega\, a}_{\Box, \mathbf{A}^u \parallel_{\text{sem}} \mathbf{B}^u} ((\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), \dot{N}_{\mathbf{A}} \cdot \dot{N}_{\mathbf{B}}).$$

We still need to show that $\dot{N}_{\mathbf{A}} \cdot \dot{N}_{\mathbf{B}} \subseteq N$. Let $\dot{\nu}_{\mathbf{A}} \cdot \dot{\nu}_{\mathbf{B}} \in \dot{N}_{\mathbf{A}} \cdot \dot{N}_{\mathbf{B}}$, then

$$\forall (V^L_{\mathbf{A}})'.\forall (V^L_{\mathbf{B}})'.(\dot{\nu}_{\mathbf{A}} \wedge \dot{\nu}_{\mathbf{B}})' \Rightarrow \psi^\circ_{\mathbf{A}} \wedge \psi^\circ_{\mathbf{B}}.$$

From (2) it follows that there exists $(s \cdot g) \in [\![V]\!]$ which satisfies $^\circ(\psi_A \wedge \psi_{\mathbf{B}})$, hence

$$\begin{aligned} \psi^\circ_{\mathbf{A}} \wedge \psi^\circ_{\mathbf{B}} &= (\exists V.\psi_{\mathbf{A}}) \wedge (\exists V.\psi_{\mathbf{B}}) \\ &= \exists V.\psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}} \\ &= (\psi_{\mathbf{A}} \wedge \psi_{\mathbf{B}})^\circ \\ &= \psi^\circ. \end{aligned}$$

Finally, it follows from (3) that $\dot{\nu}_{\mathbf{A}} \cdot \dot{\nu}_{\mathbf{B}} \in N$, which was to be shown; and clearly $((\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), (\ell'_{\mathbf{A}}, \ell'_{\mathbf{B}}), \dot{\nu}_{\mathbf{A}} \cdot \dot{\nu}_{\mathbf{B}}) \in R$ for all $\dot{\nu}_{\mathbf{A}} \cdot \dot{\nu}_{\mathbf{B}} \in \dot{N}_{\mathbf{A}} \cdot \dot{N}_{\mathbf{B}}$. □

5. Related work

The main difference to related approaches based on modal process algebra taking data states into account, e.g. [23, 24], is that they cannot naturally express logical and structural composition in the same formalism. A comparison between modal specifications and other theories such as interface automata [25] and process algebra [3] can be found in [4]. In [9], the authors introduced sociable interfaces, that is a model of I/O automata [26] equipped with data and a gxame-based semantics. Sociable Interfaces extended interface automata with a more rich synchronization scheme allowing for one-to-one, many-to-one, one-to-many and many-to-many communication as well as communication over shared variables. Sociable Interfaces where the first interface theory to encompass both communication over actions and shared variables. While their communication primitives are richer, sociable interfaces do not encompass any notion of modalities and do not have logical composition and quotient, and their refinement is based on an alternating simulation [27].

In [28] Caillaud and Raclet introduce Marked Modal Specifications in order to achieve independent implementability of reachability properties. They develop an algebra with both logical and structural composition operators which can ensure reachability properties by construction.

Transition systems enriched with predicates are used, for instance, in the approach of [29, 30] where they use symbolic transition systems (STS), but STS do not support modalities and loose data specifications as they focus more on model checking than on the (top down) development of concurrent systems by refinement.

Related work on modal interfaces, but without data can be found in [5, 31, 32].

In [19] modal I/O automata have been extended to take into account data, by adding pre- and postconditions to transition labels. Pre- and postconditions in [19] are viewed as contracts, giving rise to semantics in terms of sets of implementations. In fact, implementations with only input actions correspond to our TSD. The main difference, however, is modal refinement: [19] defines modal refinement solely on the syntactic level of modal I/O automata, rendering it incomplete and thus much coarser than modal refinement as defined here. Neither conjunction nor a quotient operation are defined in [19]; the ideas for defining a compatibility relation between communicating modal I/O automata in [19] can be easily transferred to our setting (by distinguishing between input and output actions).

In a series of works [33], Godefroid used modal specifications as an abstract representation for transition systems in a CEGAR loop process. Our abstraction technique is inspired by the one of Godefroid. The main differences are that we work with modal transition systems to represent both the specifications and the successive and refined implementations, while Godefroid works with classical state-machines and multi-valued logics for specifications. We believe that our model could be embedded in Godefroid's procedure and lead to an extension of his work. A similar observation can be made to the work of Leucker that extends Godefroid's work to games [34].

Abstraction based model checking and three valued program analysis such as presented in [7, 35] may also benefit from being revisited with the addition of abstract data in the form of MSD.

In [36] Tripakis et al. present an interface theory for systems operating in an infinite number of synchronous steps. Contracts are used to express the relationship between input and output variables of stateful synchronous components. The components are abstracted by their interfaces and one or more contracts.

One might also compare our work with approaches such as the BIP framework[37]. The BIP framework is a hierarchical work-flow for rigorous design of embedded systems, which does not consider formal verification, as we do in the form of refinement. The BIP framework considers component composition, and notably generates C code from component descriptions. An ideal combination would be a framework with refinement, quotient and conjunction that could also generate executable code from the specifications.

6. Conclusion

We have proposed a specification theory for reasoning about components with rich data state. Our formalism, based on modal transition systems, supports: refinement checking, consistency checking with pruning of inconsistent states, structural and logical composition, and a quotient operator. The specification operators are defined on the symbolic representation which allows for automatic analysis of specifications. We have also presented a predicate abstraction technique for the verification of modal refinement. We believe that this work is a significant step towards practical use of specification theories based on modal transition systems. The ability to reason about data domains permits the modeling of industrial case studies.

In the future, we intend to develop larger case studies. Furthermore, we would like to extend the formalism with more complex communication patterns, most importantly

input/output actions, and to investigate in which cases we can still obtain all the operators on specifications, in particular the quotient operator. We are also planning to implement the theory in the MIO Workbench [38, 39], a verification tool for modal input/output interfaces. The implementation in the MIO Workbench would be based on BDD [40] technology. In future work it would also be very interesting to compare the expressive power of MSD relative to Parametric Modal Transition Systems [41] and Disjunctive Modal Transition Systems (DMTSs) [12, 42]. It would also be very relevant to find infinite data domains for which the modeling and analysis will work well in practice. Also further exploring the limitations that separable transition predicates impose.

Acknowledgment. We would like to thank Rolf Hennicker for valuable comments on a draft of the conference version [1] of the paper.

References

[1] S. Bauer, K. G. Larsen, A. Legay, U. Nyman, A. Wąsowski, A modal specification theory for components with data, in: FACS:2011, Vol. 7253 of Lecture Notes in Computer Science, Springer, 2012.

[2] K. G. Larsen, Modal specifications, in: J. Sifakis (Ed.), Automatic Verification Methods for Finite State Systems, Vol. 407 of Lecture Notes in Computer Science, Springer, 1989.

[3] R. Milner, A Calculus of Communicating Systems, Vol. 92 of Lecture Notes in Computer Science, Springer, 1980.

[4] U. Nyman, Modal transition systems as the basis for interface theories and product lines, Ph.D. thesis, Department of Computer Science, Aalborg University (October 2008).

[5] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, R. Passerone, Modal interfaces: unifying interface automata and modal specifications, in: S. Chakraborty, N. Halbwachs (Eds.), EMSOFT, ACM, 2009.

[6] P. A. Abdulla, A. Bouajjani, J. d'Orso, Monotonic and downward closed games, J. Log. Comput. 18 (1) (2008) 153–169.

[7] P. Godefroid, M. Huth, R. Jagadeesan, Abstraction-based model checking using modal transition systems, in: CONCUR, Vol. 2154 of Lecture Notes in Computer Science, Springer, 2001, pp. 426–440.

[8] P. A. Abdulla, K. Cerans, B. Jonsson, Y.-K. Tsay, Algorithmic analysis of programs with well quasi-ordered domains, Inf. Comput. 160 (1-2) (2000) 109–127.

[9] L. de Alfaro, L. D. da Silva, M. Faella, A. Legay, P. Roy, M. Sorea, Sociable interfaces, in: B. Gramlich (Ed.), FroCos, Vol. 3717 of Lecture Notes in Computer Science, Springer, 2005, pp. 81–105.

[10] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, W. Yi, Uppaal - a tool suite for automatic verification of real-time systems, in: R. Alur, T. A. Henzinger, E. D. Sontag (Eds.), Hybrid Systems, Vol. 1066 of Lecture Notes in Computer Science, Springer, 1995, pp. 232–243.

[11] G. J. Holzmann, The SPIN Model Checker - primer and reference manual, Addison-Wesley, 2004.

[12] K. G. Larsen, L. Xinxin, Equation solving using modal transition systems, in: LICS, IEEE Computer Society, 1990, pp. 108–117.

[13] K. G. Larsen, B. Thomsen, A modal process logic, in: LICS, IEEE Computer Society, 1988, pp. 203–210.

[14] A. Antonik, M. Huth, K. G. Larsen, U. Nyman, A. Wasowski, Complexity of decision problems for mixed and modal specifications, in: FoSSaCS 2008, Vol. 4962 of Lecture Notes in Computer Science, Springer, 2008, pp. 112–126.

[15] C. Hoare, Proof of correctness of data representations, Acta Informatica 1 (4) (1972) 271–281. doi:10.1007/BF00289507.
URL http://dx.doi.org/10.1007/BF00289507

[16] K. G. Larsen, U. Nyman, A. Wąsowski, On modal refinement and consistency, in: L. Caires, V. T. Vasconcelos (Eds.), CONCUR, Vol. 4703 of Lecture Notes in Computer Science, Springer, 2007, pp. 105–119.

[17] N. Benes, J. Kretínský, K. G. Larsen, J. Srba, On determinism in modal transition systems, Theor. Comput. Sci. 410 (41) (2009) 4026–4043.

[18] L. de Alfaro, T. A. Henzinger, Interface theories for component-based design, in: EMSOFT, Vol. 2211 of Lecture Notes in Computer Science, Springer, 2001, pp. 148–165.

[19] S. S. Bauer, R. Hennicker, M. Wirsing, Interface theories for concurrency and data, Theor. Comput. Sci. 412 (28) (2011) 3101–3121.

[20] K. G. Larsen, U. Nyman, A. Wąsowski, Modal I/O automata for interface and product line theories, in: R. D. Nicola (Ed.), ESOP, Vol. 4421 of Lecture Notes in Computer Science, Springer, 2007, pp. 64–79.

[21] J.-B. Raclet, Residual for component specifications, Electr. Notes Theor. Comput. Sci. 215 (2008) 93–110.

[22] S. Graf, H. Saïdi, Construction of abstract state graphs with pvs, in: O. Grumberg (Ed.), CAV, Vol. 1254 of Lecture Notes in Computer Science, Springer, 1997, pp. 72–83.

[23] J. van de Pol, M. V. Espada, Modal Abstractions in μCRL, in: C. Rattray, S. Maharaj, C. Shankland (Eds.), AMAST, Vol. 3116 of Lecture Notes in Computer Science, Springer, 2004, pp. 409–425.

[24] M. V. Espada, J. van de Pol, An abstract interpretation toolkit for μCRL, Formal Methods in System Design 30 (3) (2007) 249–273.

[25] L. de Alfaro, T. A. Henzinger, Interface automata, SIGSOFT Softw. Eng. Notes 26 (2001) 109–120.

[26] N. Lynch, M. R. Tuttle, An introduction to Input/Output automata, CWI-quarterly 2 (3).

[27] R. Alur, T. A. Henzinger, O. Kupferman, M. Y. Vardi, Alternating refinement relations, in: D. Sangiorgi, R. de Simone (Eds.), CONCUR, Vol. 1466 of Lecture Notes in Computer Science, Springer, 1998, pp. 163–178.

[28] B. Caillaud, J.-B. Raclet, Ensuring reachability by design, in: A. Roychoudhury, M. D'Souza (Eds.), ICTAC, Vol. 7521 of Lecture Notes in Computer Science, Springer, 2012, pp. 213–227.

[29] F. Fernandes, J.-C. Royer, The STSLib project: Towards a formal component model based on STS, Electr. Notes Theor. Comput. Sci. 215 (2008) 131–149.

[30] T. Barros, R. Ameur-Boulifa, A. Cansado, L. Henrio, E. Madelaine, Behavioural models for distributed fractal components, Annales des Télécommunications 64 (1-2) (2009) 25–43.

[31] J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, R. Passerone, A modal interface theory for component-based design, Fundam. Inform. 108 (1-2) (2011) 119–149.

[32] G. Goessler, J.-B. Raclet, Modal contracts for component-based design, in: D. V. Hung, P. Krishnan (Eds.), SEFM, IEEE Computer Society, 2009, pp. 295–303.

[33] G. Bruns, P. Godefroid, Model checking with multi-valued logics, in: J. Díaz, J. Karhumäki, A. Lepistö, D. Sannella (Eds.), ICALP, Vol. 3142 of Lecture Notes in Computer Science, Springer, 2004, pp. 281–293.

[34] O. Grumberg, M. Lange, M. Leucker, S. Shoham, When not losing is better than winning: Abstraction and refinement for the full mu-calculus, Inf. Comput. 205 (8) (2007) 1130–1148.

[35] M. Huth, R. Jagadeesan, D. A. Schmidt, Modal transition systems: A foundation for three-valued program analysis, in: D. Sands (Ed.), ESOP, Vol. 2028 of Lecture Notes in Computer Science, Springer, 2001, pp. 155–169.

[36] S. Tripakis, B. Lickly, T. A. Henzinger, E. A. Lee, A theory of synchronous relational interfaces, ACM Trans. Program. Lang. Syst. 33 (4) (2011) 14.

[37] A. Basu, S. Bensalem, M. Bozga, J. Combaz, M. Jaber, T.-H. Nguyen, J. Sifakis, Rigorous component-based system design using the bip framework, IEEE Software 28 (3) (2011) 41–48.

[38] S. S. Bauer, P. Mayer, A. Schroeder, R. Hennicker, On Weak Modal Compatibility, Refinement, and the MIO Workbench, in: J. Esparza, R. Majumdar (Eds.), TACAS, Vol. 6015 of Lecture Notes in Computer Science, Springer, 2010, pp. 175–189.

[39] S. S. Bauer, P. Mayer, A. Legay, MIO Workbench: A Tool for Compositional Design with Modal Input/Output Interfaces, in: ATVA, Vol. 6996 of Lecture Notes in Computer Science, Springer, 2011, pp. 418–421.

[40] R. E. Bryant, Symbolic boolean manipulation with ordered binary-decision diagrams, ACM Comput. Surv. 24 (3) (1992) 293–318.

[41] N. Benes, J. Kretínský, K. G. Larsen, M. H. Møller, J. Srba, Parametric modal transition systems, in: T. Bultan, P.-A. Hsiung (Eds.), ATVA, Vol. 6996 of Lecture Notes in Computer Science, Springer, 2011, pp. 275–289.

[42] H. Fecher, H. Schmidt, Comparing disjunctive modal transition systems with an one-selecting variant, J. Log. Algebr. Program. 77 (1-2) (2008) 20–39.

Appendix A. Appendix

Some of the more trivial proofs are included in the appendix for completeness.

Proof of Thm. 6. We only consider one case, the other ones can be proven in a similar way. Assume

$$((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\Box, \langle \mathbf{A}_1 \wedge \mathbf{A}_2 \rangle_{\mathrm{sem}}} ((\ell_1', \ell_2'), S').$$

Then

$$((\ell_1, \ell_2), a, \psi, (\ell_1', \ell_2')) \in E_{\Box, \mathbf{A}_1 \wedge \mathbf{A}_2}$$

such that $S' = \{s' \in [\![V^L]\!] \mid \psi(s \cdot g, s')\} \neq \emptyset$. Then this construction must come from (w.l.o.g.) the second rule in Def. 6. Thus we have

$$(\ell_1, a, \psi_1, \ell_1') \in E_{\Box, \mathbf{A}_1}$$

and $\psi \equiv \psi_1 \wedge (\bigvee_{\psi_2 \in May_2^a(\ell_2, \ell_2')} \psi_2)$. This means that S' contains all those s' such that $\psi_1(s \cdot g, s')$ and there exists some $(\ell_2, a, \psi_2, \ell_2') \in E_{\Diamond, \mathbf{A}_2}$ such that $\psi_2(s \cdot g, s')$. Then, it also follows that

$$(\ell_2, s) \xrightarrow{g\,a}_{\Diamond, \langle \mathbf{A}_2 \rangle_{\mathrm{sem}}} (\ell_2', \{s'\})$$

for each $s' \in S'$ and moreover

$$(\ell_1, s) \xrightarrow{g\,a}_{\Box, \langle \mathbf{A}_1 \rangle_{\mathrm{sem}}} (\ell_1', S_1')$$

such that $S' \subseteq S_1'$. By semantic conjunction, we get

$$((\ell_1, \ell_2), s) \xrightarrow{g\,a}_{\Box, \langle \mathbf{A}_1 \rangle_{\mathrm{sem}} \wedge_{\mathrm{sem}} \langle \mathbf{A}_2 \rangle_{\mathrm{sem}}} ((\ell_1', \ell_2'), S'')$$

where $S'' = \{s' \in S_1' \mid (\ell_2, s) \xrightarrow{g\,a}_{\Diamond, \langle \mathbf{A}_2 \rangle_{\mathrm{sem}}} (\ell_2', \{s'\})\}$. But then $S'' = S'$ follows from maximality of S'. The other direction can be seen similarly.

□

Proof of Thm. 8. In this proof, we write $\|$ for $\|_{\mathrm{sem}}$, $\backslash\!\backslash$ for $\backslash\!\backslash_{\mathrm{sem}}$, etc. for a better readability.

$\Longrightarrow$:

We assume a relation R witnessing $\mathbf{X} \leq \rho(\mathbf{T} \backslash\!\backslash \mathbf{S})$. We define a relation $R' \subseteq (Loc_{\mathbf{S}} \times Loc_{\mathbf{X}}) \times Loc_{\mathbf{T}} \times [\![V_{\mathbf{T}}^L]\!]$ by

$$R' = \{((\ell_{\mathbf{S}}, \ell_{\mathbf{X}}), \ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \mid \exists S_{\mathbf{S}} \subseteq [\![V_{\mathbf{S}}^L]\!] : (\ell_{\mathbf{X}}, (\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \in R \text{ and } s_{\mathbf{S}} \in S_{\mathbf{S}}\}.$$

We show that R' is a refinement relation demonstrating $\mathbf{S} \parallel \mathbf{X} \leq \mathbf{T}$. First, it is easy to see that $((\ell_{\mathbf{S}}^0, \ell_{\mathbf{X}}^0), \ell_{\mathbf{T}}^0, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \in R'$ for all $(s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \in S_{\mathbf{S} \| \mathbf{X}}^0$. Now let

$$((\ell_{\mathbf{S}}, \ell_{\mathbf{X}}), \ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \in R'.$$

We can thus assume that there exists $S_{\mathbf{S}} \subseteq [\![V_{\mathbf{S}}^L]\!]$ such that

$$(\ell_{\mathbf{X}}, (\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \in R \tag{A.1}$$

and $s_{\mathbf{S}} \in S_{\mathbf{S}}$.

1. Assume $((\ell_{\mathbf{S}}, \ell_{\mathbf{X}}), s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\diamond,\mathbf{S}\|\mathbf{X}} ((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), \{s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}\})$. Then we have, by the rules of parallel composition,

$$(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond,\mathbf{S}} (\ell'_{\mathbf{S}}, \{s'_{\mathbf{S}}\}) \tag{A.2}$$

and

$$(\ell_{\mathbf{X}}, s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\diamond,\mathbf{X}} (\ell'_{\mathbf{X}}, \{s'_{\mathbf{X}}\}). \tag{A.3}$$

From (A.1) and (A.3) it follows that there exists

$$((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\diamond,\mathbf{T} \setminus\setminus \mathbf{S}} ((\ell'_{\mathbf{T}}, \ell''_{\mathbf{S}}, S'_{\mathbf{S}}), s'_{\mathbf{X}}) \tag{A.4}$$

such that $(\ell'_{\mathbf{X}}, (\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S'_{\mathbf{S}}), s'_{\mathbf{X}}) \in R$. Note that this transition cannot lead to a universal state since neither the rule $[\mathsf{data\text{-}unreachable}_{\setminus\setminus}]$ nor the rule $[\mathsf{unreachable}_{\setminus\setminus}]$ is applicable. The transition (A.4) must come from the rule $[\mathsf{may}_{\setminus\setminus}]$:

$$(\ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\diamond,\mathbf{T}} (\ell'_{\mathbf{T}}, \{s''_{\mathbf{S}} \cdot s'_{\mathbf{X}}\}) \tag{A.5}$$

and

$$(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond,\mathbf{S}} (\ell''_{\mathbf{S}}, \{s''_{\mathbf{S}}\}). \tag{A.6}$$

By determinism of $\mathbf{S}$ and (A.2),(A.6), it follows that $\ell'_{\mathbf{S}} = \ell''_{\mathbf{S}}$ and $s'_{\mathbf{S}} = s''_{\mathbf{S}}$. Thus $S_{\mathbf{S}} = \{s'_{\mathbf{S}}\}$. Finally, we can conclude that $((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), \ell'_{\mathbf{T}}, s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}) \in R'$.

2. Assume $(\ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\square,\mathbf{T}} (\ell'_{\mathbf{T}}, S_{\mathbf{T}})$. Assume that $(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \not\xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\square,\mathbf{S}}$. Then the rule $[\mathsf{error2}_{\setminus\setminus}]$ would be applicable resulting in a must-transition leading to $(\bot, \{s_{\mathbf{X}}\})$ which contradicts the fact that R is a refinement relation witnessing $\mathbf{X} \leq \rho(\mathbf{T} \setminus\setminus \mathbf{S})$. Note that any state $(\bot, S)$ is syntactically inconsistent due to rule $[\mathsf{errorstate}_{\setminus\setminus}]$. Hence there exists

$$(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\square,\mathbf{S}} (\ell'_{\mathbf{S}}, S_{\mathbf{S}}). \tag{A.7}$$

A similar argumentation shows that $T' \setminus\setminus S' = \emptyset$ cannot be the case, because $[\mathsf{error1}_{\setminus\setminus}]$ cannot be the case. So assume that $T' \setminus\setminus S' \neq \emptyset$, and by rule $[\mathsf{must}_{\setminus\setminus}]$ we have

$$((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}), S_{\mathbf{S}}), s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\square,\mathbf{T} \setminus\setminus \mathbf{S}} ((\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S_{\mathbf{S}}), S_{\mathbf{T}} \setminus\setminus S_{\mathbf{S}}). \tag{A.8}$$

By the assumption (A.1) it follows that there exists

$$(\ell_{\mathbf{X}}, s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\square,\mathbf{X}} (\ell'_{\mathbf{X}}, S_{\mathbf{X}}) \tag{A.9}$$

such that $S_{\mathbf{X}} \subseteq S_{\mathbf{T}} \setminus\setminus S_{\mathbf{S}}$ and $(\ell'_{\mathbf{X}}, (\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S_{\mathbf{S}}), s'_{\mathbf{X}}) \in R$ for all $s'_{\mathbf{X}} \in S_{\mathbf{X}}$. Parallel composition of transitions (A.7) and (A.9) yields

$$((\ell_{\mathbf{S}}, s_{\mathbf{X}}), s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}})\, a}_{\square,\mathbf{S}\|\mathbf{X}} ((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), S_{\mathbf{S}} \cdot S_{\mathbf{X}}). \tag{A.10}$$

We can conclude $((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), \ell'_{\mathbf{T}}, s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}) \in R'$ for all $(s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}) \in (S_{\mathbf{S}} \cdot S_{\mathbf{X}})$ since $(\ell'_{\mathbf{X}}, (\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S_{\mathbf{S}}), s'_{\mathbf{X}}) \in R$ and $s'_{\mathbf{S}} \in S_{\mathbf{S}}$.

$\Longleftarrow$:
We assume a relation R' witnessing $\mathbf{S} \parallel \mathbf{X} \leq \mathbf{T}$. We define a relation R by

$$R = \{(\ell_{\mathbf{X}}, (\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \mid \forall s_{\mathbf{S}} \in S_{\mathbf{S}} : ((\ell_{\mathbf{S}}, \ell_{\mathbf{X}}), \ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \in R'\}$$
$$\cup \{(\ell_{\mathbf{X}}, \mathsf{univ}, s_{\mathbf{X}}) \mid \ell_{\mathbf{X}} \in Loc_{\mathbf{X}}, s_{\mathbf{X}} \in [\![V^L_{\mathbf{X}}]\!]\}.$$

We show that R is a refinement relation demonstrating $\mathbf{X} \leq \rho(\mathbf{T} \setminus\!\setminus \mathbf{S})$. Obviously, it holds that $(\ell^0_{\mathbf{X}}, (\ell^0_{\mathbf{T}}, \ell^0_{\mathbf{S}}, S^0_{\mathbf{S}}), s_{\mathbf{X}}) \in R$ for all $s_{\mathbf{X}} \in S^0_{\mathbf{X}}$. Now, let

$$(\ell_{\mathbf{X}}, (\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \in R. \tag{A.11}$$

1. Assume $(\ell_{\mathbf{X}}, s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\diamond, \mathbf{X}} (\ell'_{\mathbf{X}}, \{s'_{\mathbf{X}}\})$.

   Subcase $s_{\mathbf{S}} \notin S_{\mathbf{S}}$**:** Then, by rule [data-unreachable$_{\setminus\!\setminus}$] we have

   $$((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\diamond, \mathbf{T} \setminus\!\setminus \mathbf{S}} (\mathsf{univ}, \{s'_{\mathbf{X}}\})$$

   and by definition, $(\ell'_{\mathbf{X}}, \mathsf{univ}, s'_{\mathbf{X}}) \in R$.

   Subcase $s_{\mathbf{S}} \in S_{\mathbf{S}}$**:**

   Subsubcase $(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \not\xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond, \mathbf{S}}$ **:** Then the rule [unreachable$_{\setminus\!\setminus}$] applies and we get

   $$((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\diamond, \mathbf{T} \setminus\!\setminus \mathbf{S}} (\mathsf{univ}, \{s'_{\mathbf{X}}\})$$

   and, as before, $(\ell'_{\mathbf{X}}, \mathsf{univ}, s'_{\mathbf{X}}) \in R$.

   Subsubcase $(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond, \mathbf{S}}$ **:** Then there exists

   $$(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond, \mathbf{S}} (\ell'_{\mathbf{S}}, \{s'_{\mathbf{S}}\})$$

   and by parallel composition we get

   $$((\ell_{\mathbf{S}}, \ell_{\mathbf{X}}), s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\diamond, \mathbf{S} \parallel \mathbf{X}} ((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), \{s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}\}).$$

   Now, from our assumption (A.11), it follows that there exists

   $$(\ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\diamond, \mathbf{T}} (\ell'_{\mathbf{T}}, \{s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}\})$$

   such that $((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), \ell'_{\mathbf{T}}, s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}) \in R'$. Finally, we have

   $$(\ell'_{\mathbf{X}}, (\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, \{s'_{\mathbf{S}}\}), s'_{\mathbf{X}}) \in R.$$

2. Assume $((\ell_{\mathbf{T}}, \ell_{\mathbf{S}}, S_{\mathbf{S}}), s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\Box, \mathbf{T} \setminus\!\setminus \mathbf{S}} p$. This must come from the rule [must$_{\setminus\!\setminus}$] (the other cases [error1$_{\setminus\!\setminus}$] and [error2$_{\setminus\!\setminus}$] lead to contradiction) implying

   $$p = ((\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S'_{\mathbf{S}}), S_{\mathbf{X}}).$$

   Hence we have

   $$(\ell_{\mathbf{T}}, s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\diamond, \mathbf{T}} (\ell'_{\mathbf{T}}, S_{\mathbf{T}})$$

and

$$(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond,\mathbf{S}} (\ell'_{\mathbf{S}}, S_{\mathbf{S}})$$

such that $S'_{\mathbf{S}} = S_{\mathbf{S}}$ and $S_{\mathbf{X}} = S_{\mathbf{T}} \setminus\!\setminus S_{\mathbf{S}}$; moreover, $s_{\mathbf{S}} \in S_{\mathbf{S}}$. From our assumption (A.11), it follows that there exists

$$((\ell_{\mathbf{S}}, \ell_{\mathbf{X}}), s_{\mathbf{S}} \cdot s_{\mathbf{X}}) \xrightarrow{g_{\mathbf{T}}\, a}_{\Box,\mathbf{S}\|\mathbf{X}} (\ell''_{\mathbf{S}}, \ell'_{\mathbf{X}}), S'_{\mathbf{S}} \cdot S_{\mathbf{X}})$$

such that $S'_{\mathbf{S}} \cdot S_{\mathbf{X}} \subseteq S_{\mathbf{T}}$ and $((\ell'_{\mathbf{S}}, \ell'_{\mathbf{X}}), \ell'_{\mathbf{T}}, s'_{\mathbf{S}} \cdot s'_{\mathbf{X}}) \in R'$ for all $s'_{\mathbf{S}} \in S'_{\mathbf{S}}$, $s'_{\mathbf{X}} \in S_{\mathbf{X}}$. Hence

$$(\ell_{\mathbf{S}}, s_{\mathbf{S}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{X}})\, a}_{\diamond,\mathbf{S}} (\ell''_{\mathbf{S}}, S'_{\mathbf{S}})$$

and

$$(\ell_{\mathbf{X}}, s_{\mathbf{X}}) \xrightarrow{(g_{\mathbf{T}} \cdot s_{\mathbf{S}})\, a}_{\diamond,\mathbf{X}} (\ell'_{\mathbf{X}}, S_{\mathbf{X}}).$$

Since $\mathbf{S}$ is deterministic, we have $\ell'_{\mathbf{S}} = \ell''_{\mathbf{S}}$ and $S_{\mathbf{S}} = S'_{\mathbf{S}}$. $S_{\mathbf{X}} \subseteq S_{\mathbf{T}} \setminus\!\setminus S_{\mathbf{S}}$ since we know $S_{\mathbf{S}} \cdot S_{\mathbf{X}} \subseteq S_{\mathbf{T}}$. Finally, it is easy to see that, for all $s'_{\mathbf{X}} \in S_{\mathbf{X}}$,

$$(\ell'_{\mathbf{X}}, (\ell'_{\mathbf{T}}, \ell'_{\mathbf{S}}, S_{\mathbf{S}}), s'_{\mathbf{X}}) \in R.$$

□

Proof of Thm. 9. We will show it for must-transitions only, for may-transitions the claim can be shown in a similar and straightforward way – note that each rule in Def. 7 corresponds to exactly one rule in the definition of the semantic quotient (see page 20).

We have to prove that there is a (reachable) must-transition

$$((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), q) \xrightarrow{(g \cdot s)\, a}_{\Box,\langle \mathbf{B} \setminus\!\setminus \mathbf{A}\rangle_{\mathrm{sem}}} ((\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, \xi'_{\mathbf{A}}), Q')$$

in $\langle \mathbf{B} \setminus\!\setminus \mathbf{A}\rangle_{\mathrm{sem}}$ if and only if there is a (reachable) must-transition

$$((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, [\![\xi_{\mathbf{A}}]\!]), q) \xrightarrow{(g \cdot s)\, a}_{\Box,\langle \mathbf{B}\rangle_{\mathrm{sem}} \setminus\!\setminus_{\mathrm{sem}} \langle \mathbf{A}\rangle_{\mathrm{sem}}} ((\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, [\![\xi'_{\mathbf{A}}]\!]), Q')$$

in $\langle \mathbf{B}\rangle_{\mathrm{sem}} \setminus\!\setminus_{\mathrm{sem}} \langle \mathbf{A}\rangle_{\mathrm{sem}}$.

Assume a reachable state $(\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}})$ in $\langle \mathbf{B} \setminus\!\setminus \mathbf{A}\rangle_{\mathrm{sem}}$, and assume

$$((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), q) \xrightarrow{(g \cdot s)\, a}_{\Box,\langle \mathbf{B} \setminus\!\setminus \mathbf{A}\rangle_{\mathrm{sem}}} ((\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, \xi'_{\mathbf{A}}), Q'). \tag{A.12}$$

This transition must come from a symbolic transition

$$((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, \xi_{\mathbf{A}}), a, \psi, (\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, \xi'_{\mathbf{A}})) \in E_{\Box,\mathbf{B}\setminus\!\setminus\mathbf{A}}$$

and $Q' = \{q' \in [\![V^L_{\mathbf{B}\setminus\!\setminus\mathbf{A}}]\!] \mid \psi(q \cdot g \cdot s, q')\} \neq \emptyset$. This symbolic transition must be generated by the second rule of Def. 7, thus we have

$$(\ell_{\mathbf{B}}, a, \psi_{\mathbf{B}}, \ell'_{\mathbf{B}}) \in E_{\Box,\mathbf{B}} \text{ and } (\ell_{\mathbf{A}}, a, \psi_{\mathbf{A}}, \ell'_{\mathbf{A}}) \in E_{\Box,\mathbf{A}}$$

such that $\psi \equiv \xi_{\mathbf{A}} \wedge {}^\circ\psi_{\mathbf{B}} \wedge {}^\circ\psi_{\mathbf{A}} \wedge (\psi^\circ_{\mathbf{B}} \setminus\!\setminus \psi^\circ_{\mathbf{A}})$,and $\xi'_{\mathbf{A}} \equiv \psi^\circ_{\mathbf{A}}$. Then we have

$$(\ell_{\mathbf{B}}, s \cdot q) \xrightarrow{g\, a}_{\Box,\langle B\rangle_{\mathrm{sem}}} (\ell'_{\mathbf{B}}, [\![\psi^\circ_{\mathbf{B}}]\!]) \text{ and } (\ell_{\mathbf{A}}, s) \xrightarrow{(q \cdot g)\, a}_{\Box,\langle \mathbf{A}\rangle_{\mathrm{sem}}} (\ell'_{\mathbf{A}}, [\![\psi^\circ_{\mathbf{A}}]\!])$$

By the definition of Q', it follows that

$$\begin{aligned} Q' &= \{q' \in [\![V^L_{\mathbf{B} \between \mathbf{A}}]\!] \mid \psi(q \cdot g \cdot s, q')\} \\ &= \{q' \in [\![V^L_{\mathbf{B} \between \mathbf{A}}]\!] \mid (\psi^\circ_{\mathbf{B}} \between \psi^\circ_{\mathbf{A}})(q')\} \\ &= [\![\psi^\circ_{\mathbf{B}} \between \psi^\circ_{\mathbf{A}}]\!] \\ &= [\![\psi^\circ_{\mathbf{B}}]\!] \between [\![\psi^\circ_{\mathbf{A}}]\!]. \end{aligned}$$

By rule $[\mathsf{must}_{\between}]$ (remember that $Q' \neq \emptyset$) and by the fact that $\xi_{\mathbf{A}}(s)$, we get

$$((\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, [\![\xi_{\mathbf{A}}]\!]), q) \xrightarrow{(g \cdot s)\, a}_{\square, \langle \mathbf{B} \rangle_{\mathrm{sem}} \between_{\mathrm{sem}} \langle \mathbf{A} \rangle_{\mathrm{sem}}} ((\ell'_{\mathbf{B}}, \ell'_{\mathbf{A}}, [\![\psi^\circ_{\mathbf{A}}]\!]), Q').$$

For the reverse direction, observe that for any given reachable location $(\ell_{\mathbf{B}}, \ell_{\mathbf{A}}, S_{\mathbf{A}})$ in $\langle \mathbf{B} \rangle_{\mathrm{sem}} \between_{\mathrm{sem}} \langle \mathbf{A} \rangle_{\mathrm{sem}}$ we know that $S_{\mathbf{A}}$ must be described by either the initial predicate $\varphi^0_{\mathbf{A}}$ or by $\psi^\circ_{\mathbf{A}}$ for some transition predicate $\psi_{\mathbf{A}}$ occurring in $\mathbf{A}$.

□